Development of Computational Intelligence Methods to Deal with Classi cation Problems

Ekong, Udeme Essien

Awarding institution:
King's College London

Development of Computational Intelligence Methods to Deal with Classification Problems

Udeme Essien Ekong

Department of Informatics

King's College London

A thesis submitted for the degree of

Doctor of Philosophy

September 2015

Acknowledgements

The completion of this thesis was made possible through the effort and help of numerous persons who will be mentioned in the succeeding paragraphs. I am eternally grateful to these individuals and words alone cannot fully express my level of gratitude. The greatest influence on the work done in this thesis has been my supervisor, Dr Hak-Keung Lam who has been truly instrumental in terms of the help provided and ideas shared in being able to carry out the research conducted. His immense ability and passion for the field of computational intelligence has been a great inspiration for me and I am forever grateful to him for the help and support throughout the duration of this PhD.

My progress up to this stage would have been impossible without my wonderful parents, Dr and Mrs Ekong as they have been a massive inspiration, encouragement and example to me right from infancy up until now and this has been very instrumental in my achievements up until this moment. I would also like to thank my Uncle Pat and Aunty Emem who have been a constant encouragement onto me also. My family as a whole has been absolutely supportive and I am deeply grateful for this.

Many thanks also go to Dr Helge Wurdemann and Dr Yohan Noh for their support and advice and also the encouragement during times of frustration. I would also like to thank Dr Kasper Althoefer for his support and advice at critical and transitory moments. During the PhD, I have worked closely with a number of colleagues in the research conducted and I am greatly thankful to Kelvin Bo especially for his availability and collaboration during the work conducted.

The PhD would not have been possible without the funding provided by the Niger Delta Development Committee (NDDC) who have been prompt with payment and both responsive and supportive.

Finally I would like to thank God, my creator and heavenly father for without him I am nothing.

Abstract

In the thesis presented here, variations of two very prominent machine learning techniques, the Neural Network (NN) and Support Vector Machine (SVM) are used in an attempt to solve two classification problems. Classification involves the assignment of an unknown object into a pre-determined group which consists of a set of pre-classified objects with similar features to that unknown object. The main theme of the research conducted in this thesis involves investigation into existing and proposed classifier architectures to improve the classification performance for certain research problems. The aim of the research conducted is to develop new classifiers that are robust and able to show a high level of classification accuracy to the problems that are being considered.

The problems being considered in this thesis are material surface classification and epilepsy seizure phase classification. The material surface classification problem involves the classification of a material based on its surface features which are obtained from a tactile-sensing robotic arm. Feature extraction is carried out on this input and the classifier is then used to classify based on the extracted feature inputs. Epileptic seizure is a common neurological disorder which causes the sudden discharge of cortical neurons in the brain. This results in the

onset of seizures lasting from a few seconds to around a minute. The input consists of data obtained from the electroencephalograph (EEG) of patients who suffer from epilepsy. The input is then subjected to feature extraction and the extracted feature inputs are applied to the classifier. Four traditional classifiers, namely SVM, NN, k-nearest neighbour (kNN) and naive Bayes classifier are utilised for comparison purposes to evaluate the performance of the proposed classifiers during the research conducted. To evaluate the robustness property of the classifier, the original data is contaminated with Gaussian white noise at various levels. The results of the research carried out are presented in three parts :

- The performance of six commonly used neural-network-based classifiers are investigated in solving the material surface classification problem. The significant contribution from the research conducted in this section is in the application of the neural network architectures to a novel problem (i.e material classification). The neural network architectures are also altered and re-structured in order to fit the problem space. Experimental results show that the parallel-structured, tree-structured and naive Bayes classifier outperform the others based on the average classification accuracy of the classifier when under the original data. The tree-structured classifier demonstrates the best robustness property under the noisy data.
- In continuation of the research conducted in the previous sec-

tion, a novel neural network having variable weights is proposed to deal with the material classification problem. The aim of doing this is to compare its performance to the best out of the 6 neural network architectures applied in dealing with the material classification problem. The epilepsy seizure phase classification problem is also introduced with the proposed variable weight neural network being implemented to deal with this problem. It is shown that the variable weight neural network (VWNN) classifier outperforms the traditional methods in terms of classification accuracy and robustness property when the input data is contaminated with noise.

- A novel Interval Type-2 Fuzzy Support Vector Machine (IT2FSVM) classifier has been proposed to deal with the epilepsy seizure phase classification problem. The performance of the classifier is measured based on its classification accuracy for each of the epilepsy phases. Three traditional classifiers (SVM, kNN and naive Bayes) are used for comparison purposes. The results obtained from simulations show that the novel IT2FSVM is able to show improved performance in terms of the average classification accuracy when compared to the other classifiers under the original dataset and also shows a high level of robustness when compared to other classifiers under a noisy dataset.

Statement of contributions

According to the knowledge of the author, the following contributions are asserted to be original:

- Study of neural-network based classifiers and its application to a material surface classification problem which is reported in chapter 3. Six well known neural network architectures (one-against-all, weighted one-against-all, binary coded, parallel structured, weighted parallel-structured and tree-structured) were introduced to deal with the material surface classification problem. The classifiers are judged based on their classification accuracy and robustness to a noisy input.
- VWNN and its application on material surface and epilepsy seizure phase classifications which is reported in chapter 4. The VWNN allows its weights to be changed based on the inputs to the network. A novel VWNN is proposed where the weights at each of the hidden layer levels of the tuned neural network is supplied by the tuning neural network.
- A novel IT2FSVM is proposed and applied to an epilepsy seizure phase classification problem as is reported in chapter 5. The IT2FSVM has a membership function which is shaped with the

aid of a genetic algorithm with the final output of the classifier determined by a majority rule voting system.

List of Publications

Journals

1. H.K. Lam, U. Ekong, H. Liu , B. Xiao, H. Araujo, S.H. Ling and K.Y. Chan, "A study of neural-network-based classifiers for material classification", Neurocomputing, 2014, Volume 144, p. 367-377.

2. H.K. Lam, U. Ekong, B. Xiao, G. Ouyang, H. Liu, S.H. Ling and K.Y. Chan, "Variable weight neural networks and their applications on material surface and epilepsy seizure phase classifications", Neurocomputing, 2015, Volume 149, p. 1177-1187.

3. U. Ekong, H.K. Lam, B. Xiao, G. Ouyang, X. Yin, H. Liu, S.H. Ling and K.Y. Chan, "Classification of epilepsy seizure phase using interval type-2 fuzzy support vector machines", Computing, September 2015 (Submitted).

Contents

List of Figures

List of Tables

1. AHP - Analytical hierarchy process
2. BP - Backpropagation
3. DTW - Discrete time warping
4. DWT - Discrete wave transform
5. EBP - Error backpropagation
6. ECG - Electrocardiogram
7. EEG - Electroencephalogram
8. ELM - Extreme learning machine
9. FOU - Footprint of uncertainty
10. FSVM - Fuzzy support vector machine
11. GA - Genetic algorithm
12. IT2FIS - Interval type-2 fuzzy inference system
13. IT2FSVM - Interval type-2 fuzzy support vector machine
14. KM - Karnik-Mendel
15. kNN - k-Nearest neighbour
16. LMF - Lower membership function
17. NFN - Neural fuzzy network
18. NN - Neural network

19. NNLS - Neural network with link switches
20. NSC - Network switch controller
21. NSVM - Network-based support vector machine
22. PCA - Principal component analysis
23. RBF - Radial basis function
24. RELM - Regularised extreme learning machine
25. RLS - Regularised least-squares
26. SFM - Support feature machine
27. SVM - Support vector machine
28. SVR - Support vector regression
29. SWD - Spike-and-wave discharges
30. UMF - Upper membership function
31. VWNN - Variable weight neural network

Chapter 1

Introduction

1.1 Introduction

Classification is a process that takes samples from objects and assigns each one of them to a pre-defined group or class label. In general, a classification process usually consists of three main phases. In the first phase, data from the objects have to be collected for the design of classifiers. In the second phase, feature extraction is performed to extract characteristics from the collected data to be classified such that redundant information is removed and representative information is extracted resulting in reduction of input dimensions and improved classification accuracy. In the third phase, a classifier is designed using the extracted feature data. Existing classification techniques include traditional and machine learning methods. An example of traditional methods are linear discriminant analysis [4], logic based method (e.g., decision trees [136]), statistical approach (e.g. Bayesian classification [127]) and instance-based methods (e.g. nearest neighbor algorithm [148, 151]). Machine learning methods include the support vector machines and neural network [8, 41, 70, 130]. These methods are discussed in the literature review section of the thesis. Classification techniques have been applied to solve a wide range of problems e.g. classification of different investments and lending opportunities as acceptable or unacceptable risk [155], classification of electrocardiogram (ECG) arrhythmias [102], classification of ECG beat [25], facial recognition [41, 86], hand-writing recognition [18, 69, 70, 71, 108], heart sound classification [24], human body posture classification [58], speaker verification [8], speech recognition [21, 72] and text classification [93, 130]. The two classification problems that are investigated in this thesis are that of material surface classification and epilepsy seizure phase classification.

Material classification is a very relevant field which involves the development of algorithms and classifier architectures to aid in the classification of materials via a supervised learning method. The classifier is designed and trained with a labelled input dataset and is then applied to unseen data with the aim of maximising the classification accuracy. The ability to classify materials is of vital importance to the way we human beings view our surroundings since it enhances our understanding of the world and enables us to better interact with the objects around us. Given the ever present nature of materials, a robust material classification system would be beneficial for a wide variety of applications such as in quality assurance [55], selection of appropriate material for construction work [7, 149] and failure analysis [124]. The ability to accurately differentiate between certain materials is therefore a very important field of study and as a result there is a significant amount of research work that has already been conducted in this area.

Material classification has very important and in some cases life-saving applications that serve to underline the importance and relevance of the research being conducted in this field of study. Life-saving applications include passenger safety where material classification is applied in the monitoring of the condition of railway tracks [37]. It is also applied in war zones where the photon scattering method is used to detect hidden explosive devices in the sand [112], another vital life-saving application is in rescue missions from wreckage sights [112] where material classification is used to significantly reduce the time required to identify and remove humans from the wreckage. Quality control is another important application of material classification such as in the detection of foreign objects in a manufacturing plant for cigarettes [109]. This is a very vital field in many industrial sectors and it is imperative that unwanted materials be detected and discarded from the processing chain. Material classification is also used in tactile systems to mimic the functionality of the human hand. Two main activities that benefit from tactile sensing are: humans interpreting tactile signals and using prior experience to classify materials and textures only by touching surfaces, and also in the grasping of objects by applying just enough grip force to hold objects without slipping. Both tasks are very important for a robot if we are to operate autonomously in unstructured environments [16].

The material classification problem can be split into three distinct phases which pose different levels of difficulty in implementation. In the first phase, we device a method that is used for obtaining the characteristics of the materials (e.g frictional coefficient, frequency and damping ratio [124]), a potential challenge with this phase is in selecting the appropriate method for obtaining the material characteristics and also the inherent noise that comes as a consequence of the measuring equipment. In the second phase, feature extraction is carried out in order to extract the relevant data from the raw input data and also prevent redundancies, this helps in reducing the complexity and computational costs of the classifier. The dilemma faced in this stage is in selecting the appropriate feature extraction methods in order to select the best features and therefore improve the classification accuracy by reducing the dimensional complexity of the input. In the third phase, a classification technique is used to classify the different materials based on the feature inputs, this is the most important phase and a number of classification methods have been applied in finding a solution to this problem, the objective is to find a robust classifier that is able to provide significant improvement in the classification accuracy. The objective of the research being carried out in material classification is to introduce and implement novel classifiers and classification architectures that are able to significantly improve the classification performance with regards to existing problems and also to implement classifiers to newer problems by adjusting their architecture to fit the problems pace.

The second problem that is addressed in the research conducted is that of epileptic seizure phase classification. Epilepsy, which is characterized by its ability to instantiate recurrent seizures (an interruption of normal brain functions) which are unforeseen in nature is a very common and significant neurological disorder caused by a sudden discharge of cortical neurons [30, 82]. The unexpected nature of these seizures has proven to have an adverse effect on the quality of life for those who are suffering from them. The impact is most prevalent in the formative stages of a childs life as we see an increase in the requirements for special education and also a higher incidence of below-average school performance [62, 82] in children who suffer from this condition. It also proves life-threatening in situations where the sufferer is isolated at the time of its occurrence and there is no experienced or medical help on hand to alleviate the situation. 25% of the worlds 50 million people with epilepsy have seizures that cannot be controlled by any available treatment. The need for new therapies and success of similar devices to treat cardiac arrhythmias, has spawned an explosion of research into algorithms for use in implantable therapeutic devices for epilepsy [119]. Therefore having an accurate understanding or predictive model for the pre-seizure phase (the transition towards an absence seizure occurrence) is a very vital task as it would provide the sufferers and their carers with enough notice of the upcoming seizure so they could prepare themselves and dampen the impact of the seizure occurrence. The electroencephalogram (EEG) signal is used for the purpose of epileptic detection as it is a condition related to the brains activity [121]. It consists of a recording of the electric potentials generated by the brain. Typically sixteen channels of data are recorded by measuring the potential difference between pairs of electrodes placed on the scalp. EEG is an important clinical tool for diagnosing, monitoring and managing neurological disorders related to epilepsy. The presence of epileptiform activity in the EEG confirms the diagnosis of epilepsy, which is sometimes confused with other disorders producing similar seizure-like activity. Between seizures, the EEG of a patient with epilepsy may be characterised by occasional epileptic form transients-spikes and sharp waves. Most of the existing algorithms focus on either detecting unequivocal EEG onset of seizures or on quantitative methods for predicting seizures in the state space, time, or frequency domains that may be difficult to relate to the neurophysiology of epilepsy [119].

The classification of epilepsy risk levels is difficult due to the fact that individual laboratory findings and symptoms are often inconclusive [121]. As the epileptic seizure does not occur periodically and is therefore unpredictable, it is necessary to record the EEG signal of a patient over several days, the detection by an encephalographer would therefore be time consuming due to the length of EEG recording as it would be required that they scan the entire length of the EEG recording in search of spikes and seizures [106]. Another difficulty is that different specialist may come up with differing conclusions [1]. One of the main aims of epileptic seizure research is to help the encephalographer in the time consuming task of epilepsy detection [107]. For each and every recording of the EEG, the encephalographer always attempts to understand and determine the true nature and exact locations of the EEG patterns. If this process were to be automated, it would save time, solve the subjectivity problem and try to obtain a better and more accurate diagnosis. This would then enable future works in the field to be conducted at greater speed [1]. This is a significant motivation for the research that has been conducted here [106].

The epilepsy seizure phase classification process of EEG signals consists of three main phases which are data collection, feature extraction and classification. In the first phase we have data collection where an EEG system with 10-20 electrode placement is used to obtain the raw EEG input data. One of the classic problems/difficulties associated in this phase is the noisy nature of the data and also the level of precision of the measuring instrument. In the second phase we have feature extraction which involves converting the raw EEG signal into a more efficient input dataset that results in improved classification accuracy. The challenge here is in being able to select the appropriate features that have a significant effect on classification results whilst also getting rid of redundant information from the input EEG data. The third phase involves the application of a classifier to classify the feature extracted inputs to a desired level of accuracy, this is the most important phase of the classification process as the ability to select an appropriate classifier would have a significant effect on the classification accuracy. A lot of research has been carried out into differentiating between the various seizure phases (seizure-free, pre-seizure and seizure) and these are reviewed in the following section. However, the main aim of the research into this problem is to be able to accurately classify the pre-seizure phase which would enable the patient to be adequately prepared for the onset of an epileptic seizure. The objective of the research being conducted here is to review existing classification methods that have been applied to epilepsy seizure classification and also propose novel methods that are able to show an improved performance in comparison to these methods.

The research carried out in this thesis is motivated by these two research problems with the aim of proposing and investigating various classification techniques to be able to provide suitable and robust classifiers to solve these classification problems.

1.2 Aims and Organisation

The aim of this thesis is to provide competent and robust classifiers based on computational intelligence which are applied to two major problems. These are the material surface classification and epilepsy seizure phase classification problems. The motivation of the research carried out in this thesis is based on the need for more robust classifiers and also the fact that a relatively small amount of research has been carried out into this particular field. This therefore leaves a large scope for discovery into these particular areas of research. The thesis is structured as follows:

- In chapter 2, background knowledge of the support vector machine (SVM), neural network (NN) and interval type-2 fuzzy inference system (IT2FIS) is provided.

- In chapter 3 the performance of six commonly used neural-network-based classifiers (one-against-all, weighted one-against-all, binary coded, parallel-structured, weighted parallel-structured and tree-structured) are investigated in solving the material surface classification problem which aims to identify the object nature based on surface features of the object.

- In chapter 4, a novel neural network having variable weights which is able to improve its learning and generalization capabilities is proposed to deal with the epilepsy seizure phase classification and material classification problems. The VWNN allows its weights to be changed in operation according to the characteristic of the network inputs.

- In chapter 5, an IT2FSVM classifier is proposed to deal with the epilepsy seizure phase classification problem. The performance of the classifier is measured based on its classification accuracy for each of the epilepsy phases.

- In chapter 6, a conclusion to all the proposed methods adopted and research conducted in the thesis is drawn.

Chapter 2

Background Knowledge

2.1 Literature Review

There has been a significant amount of research carried out into classification with a vast number of classifiers and also hybrid classifiers being used. This section will be organised as follows: I will first review the work done in the two research problems investigated in this thesis which are the material surface classification problem and the epilepsy seizure phase classification problem. Then I will review relevant paper regarding classification techniques.

2.1.1 Material Surface Classification

From the review of existing research, there are a number of methods to obtain the characteristic data of the materials. These include tactile sensing [15, 23, 55], hyperspectral imaging [7, 32], modal analysis [124], polarimetric imaging [51, 129] and photon scattering [112]. Tactile sensing is a method based on the artificial sense of touch implemented in robotics. The sensors used are surface sensors which are attached to the surface of the robotic fingers. The robotic finger would then come in contact with the surface by either sliding on it, or bouncing on the surface like in [15] and the surface coefficients are recorded. Hyperspectral imaging is unlike other similar imaging techniques where every pixel of the image is mapped onto one of the reference spectra, this method uses the data itself to create clusters of pixels with the same material. Hyperspectral imaging[7] involves collecting image data simulataneously in several spectral bands and therefore making it possible to obtain a continuous spectrum for each image cell. The image cell or pixel would then be a vector of n different values which are each representing a different spectral band. Classification is done by matching each input image spectrum individually to one of the reference spectra from the spectral library. The matching is done using some measure of the goodness of fit, with the best match being labelled as the winner and given the classification label.

Another method for obtaining the characteristic data of materials is modal analysis which involves the study of the dynamic characteristics of a material induced by a vibrational excitation. Under this excitation, three parameters (frequency, damping ratio and mode shape) are obtained. This constitutes the input data to the classifier. In [124] we see modal analysis being combined with the neural network in order to classify between glass and stainless steel. Polarimetric imaging [51] is another method for obtaining the characteristic data of the materials which is based on the concept of polarization. Polarization is a property of light or electromagnetic radiation that conveys information about the orientation of the transverse electric and magnetic fields. This complements other electromagnetic radiation attributes such as intensity and frequency. This therefore makes polarization a useful tool in material classification. The polarimetric imaging method is built on an iterative model which aims to recover the complex index of refraction of a specular target from multiple polarization measurements. The recovered parameters would then be used to discriminate between objects with the aid of the nearest neighbour rule. A relatively new approach towards obtaining characteristic data of materials is that of photon scattering. In [112] the scatter components of an interrogating gamma-ray radiation beam was used in order to determine the types of material that have been embedded in sands and also to determine the depth of the material. The scattered protons were collected using a planar surface detector located directly above the sample. The collected signals were then subjected to feature extraction in the frequency domain before being used as the input to the classifier.

The classification methods used include the neural network [15, 32, 112, 124, 149], naive Bayes [55], hyperspectral image classifier [7], SVM [23], regularized least squares (RLS) [23], regularized extreme learning machine (RELM) [23] and kNN [51, 129]. In [149] we observe a hybrid classifier which consists of the neural network and the GA being used for classification. This method involves the optimization of the interconnection weights. The GA is therefore being used to improve/optimize the backpropagation algorithm and therefore improve the performance of the classifier.

The existing literature poses some relevant and insightful research into this area but we see that there are some aspects that are left wanting. For example we do not see an adequate method which shows a high level of flexibility in being able to add or subtract material classes from the classifier, also the classification accuracy obtained by some of the existing methods [112] offer room for improvement. The research carried out in this thesis aims to address these issues by proposing classifiers that are able to meet these criterion whilst also posting improved levels of robustness and classification accuracy.

2.1.2 Epilepsy Seizure Phase Classification

As mentioned in the previous section, this problem contains three phases with phase 1 involving the obtaining of input EEG data, phase 2 being feature extraction and phase 3 involving the classification of feature extracted inputs. As this is a time-series input signal, feature extraction can be carried out on 3 levels (features related to the frequency, features related to time and the wavelet). A number of feature extraction methods have been applied to epileptic seizure research, these include principal component analysis (PCA) in [36], empirical mode decomposition in [59], discrete wavelet transform (DWT) in [103], the lifting based discrete wavelet transform in [26, 135] and bivariate feature extraction in [19]. The basic idea of the DWT is to represent the time-series input as a linear combination of wavelet basis functions, keeping only the first N coefficients. The wavelet transform represents an arbitrary function $f(t)$ as a superposition of a set of wavelets or basis functions. The basis functions are obtained from a single prototype wavelet or the mother wavelet by shifting and scaling. The lifting based discrete wavelet transform is a variation of the DWT where the decomposition is done via three lifting steps, the first step involves converting the wavelet filters into a polyphase matrix, then a Euclidean algorithm is applied to factor the polyphase matrix into elementary matrices, the elements of the decomposed matrices are then used as coefficients.

A number of traditional and hybrid classifiers have been applied to classify the extracted features from the EEG signal such as the neural network [2, 26, 34, 36, 94, 104, 135], support vector machine [13, 19, 103], kNN [12], support feature machine (SFM) [13] and extreme learning machine (ELM) [34]. In [36] a method which uses the PCA feature extraction method and the cosine RBF NN is proposed. The results show that inclusion of the PCA feature extraction method has a positive effect on the classification performance of the classifier as the classifier has a significantly higher level of classification with the feature input when compared to classifying with the raw EEG input. In [12] a novel time-series method is proposed which involves the integration of the time-series similarity measure with the kNN classifier. The classification method involves three key phases. In the first phase, a quantitative measure of the brain dynamics is obtained. This is done with the aid of short-term maximum Lyapunov exponent. The second phase is based on the statistical comparison for similarity/dissimilarity of classifiable features of the different seizure states, the time-series similarity measures that are applied are the Euclidean, t-statistical and dynamic time warping (DTW) distances. The third phase involves the classification algorithm which is based on the kNN classifier.

In [13] two classifiers are proposed, the first is the support feature machine (SFM) which employs the nearest neighbour rule and time-similarity measures in order to classify the EEG signals. Its optimization model selects features with strong class separability in order to maximise the classification accuracy. Secondly a network-based version of the support vector machine (NSVM) is proposed where statistical and correlation measures among time-series profiles are incorporated. The NSVM overcomes one of the drawbacks of the traditional SVM for time-series analysis as the traditional SVM treats each time stamp of a time-series as an independent variable whereas the data are actually highly correlated. In [34] we see the EEG signals being classified with the aid of a hybrid extreme learning machine working in conjunction with a feed-forward neural network. The initial weights of the feed-forward NN are selected with the aid of the analytical hierarchy process (AHP) method, the output weights of the NN are analytically selected with the aid of the ELM. In [44, 45, 46, 120, 122, 123] we see extensive research being carried out into a fuzzy logic application into epileptic seizure classification. The fuzzy logic approach is used to evaluate the epilepsy risk levels of patients from the EEG signals, with the majority of research carried out involving the optimization of the fuzzy outputs. Optimization techniques that are covered include the decision trees, binary and continuous genetic algorithm and the minimum relative entropy method.

From the existing research we see that there is a lack of emphasis on the classification of the pre-seizure stage of epilepsy onset (the most important stage to recognize in enabling patients to live better with the condition as they are able to prepare themselves). In some of the research carried out [2, 12, 94] the classifier is built to separate between the seizure-free and seizure phases with little research carried out into the pre-seizure phases. We also see a relatively poor level of classification accuracy obtained by the existing classifiers in separating between the seizure phases [2], with little or no tests on the robustness of said classifiers. The research carried out in this thesis aims to address these issues by proposing a classifier that is able to distinguish between the seizure-free, pre-seizure and seizure phases and also providing a high level of robustness and classification accuracy.

2.1.3 Traditional Classification Methods

Bayesian decision theory [127] is one of most important methods in statistical classification which offers a primary model for further classification procedures. Naive Bayes classifier is based on the assumption that equal prior probabilities exists for all classes [67]. This reduces the complexity of analysis and helps in resolving conflicts that occur when two or more classes are not well separable, resulting in improving the classification accuracy. Although Bayesian decision is simple and powerful, the posterior probabilities cannot be determined directly [147]. The Bayesian classifier has also been successfully implemented to many real-world applications e.g. weeds identification [127], medical diagnosis [146], speech recognition [92], image classification [133], credit score modelling [144].

Another traditional classification technique is the k-nearest neighbour technique (kNN) [151] which is easy to apply and good in dealing with text based problems such as visual category recognition [148]. However, the kNN has its intrinsic limitations with the main disadvantages being the large memory requirements and also the lack of a logical way to choose the best value for "k". This would therefore introduce difficulties to a classification application as different data sets require an optimized value of "k" in order to improve the performance of this method [66]. Furthermore, the precision accuracy of kNN declines when there are too many classes to deal with or when an uneven density of training samples is presented.

2.1.4 Machine Learning Methods

Machine learning methods include single layer perceptron [88], artificial neural networks [31, 69, 147], support vector machines and neural-fuzzy networks [25, 58, 71, 72, 102]. The first neural networks were designed based on mathematics and algorithms in the 1940s. The McCulloch-Pitts neuron was proposed in 1943, this laid the foundation of modern neural networks [31]. However, there was no effective neural network training algorithm, so the development of neural networks was stagnated for some years. After that, a trainable network with adaptive elements, which are the building blocks, was designed [31]. A single layer perceptron [88] neural network model was first introduced by Rosenblatt in 1962, which cast a huge impact on the artificial intelligence field. Since then, different types of perceptron-based techniques have emerged in large numbers.

A single layer perceptron has a simple structure which can be seen as a component that adds weights to the inputs and then computes the sum to the output of the system. After that, the outputs are used to compare with the corresponding targets to verify the accuracy. With the information of difference between outputs and targets, the weights can be adjusted to achieve higher level of accuracy. However, there is a major limitation that restrains the applications, only linearly separable problems can be solved by the single layer perceptron. Although it has the major limitation, the single layer perceptron has also been implemented well to applications such as finger print matching [88] and image detection [111].

2.1.5 Neural Network

The neural network has been applied into numerous areas such as surface texture classification [137], aerial image classification [39], handwriting classification [63], classification of ground vehicles [60], leaf classification [33], fault diagnosis [73] and epilepsy detection [65]. In [53] we see the NN architecture being used to implement a kNN classifier. This NN architecture uses a k-Maximum network as opposed to the winner-take-all method used by the standard NN and other techniques in order to select the maximum input. The significant issue is that under this method, the NN does not need to be trained further after the initial weights are set.

A number of hybrid classifiers have also been proposed in the literature such as in [87] where the NN is merged with the GA for the weight training phase of the neural network. This method is carried out in 3 phases, the first phase involves the representation of the connection weights either as a binary string or a real number. The second phase occurs as the evaluation of the connection weights fitness. This is done by constructing the corresponding NN and computing its fitness function and mean square error (MSE) function. The third and final phase involves the application of an evolutionary process like crossover or mutation according to the fitness from phase 2. The iterative evolution would stop when the fitness is greater than a predefined value (i.e the training error is smaller than a certain value) or the population has converged.

The research carried out in [73] also has a hybrid classifier which is based on the merger of the NN with GA and wavelet transform. The NN has the limitation when using the BP algorithm of only being able to find the local solution to the weight optimization problem but the GA is able to find the global solution and is therefore complementary to the NN. In the reviewed research, the wavelet transform is applied as a superior feature extraction method when compared to traditional methods like the fourier transform. The initial weights of the network are determined automatically using the GA instead of random selection. The global solution is then found by the NN using the BP algorithm in this small solution space.

2.1.5.1 Variable Structure Neural Network

The variable structure neural network is a very interesting and successful field of study with applications in areas such as the prediction of electricity demand [29, 42, 80], prediction of animal phenotype values [77], control systems [11, 83, 113, 141, 154], Pattern recognition [81], fuel cell model [105], facial recognition [6], chaos reproduction [28] and stability analysis [17, 150]. In the literature, the structure of the neural network is varied using components such as the synaptic weights [28, 29, 57, 95], basis functions in the RBF NN [11, 83, 90, 105], hidden neurons [57, 77, 95, 154], switch controller [80, 81], the learning rate [6] and by adding/subtracting delays from the network [17, 145, 150].

For the VWNN there are a number of different methods being used to optimize the weights such as the least squares method in [29]. In [28] the structure of the network is changed based on the output performance. An initial configuration is first assumed, if a pre-specified error bound is not reached, more synaptic connections will be added to the network and another iteration is carried out. This process is repeated until the output performance satisfies the pre-set error bound or a maximum number of iterations is reached.

The RBF NN is also prominent in the literature with a number of ways being proposed to determine how the number of basis functions in the network are altered. In [90, 105] the number of basis functions is increased or decreased based on the approximation error of the network relative to the desired level of accuracy with the aid of a self-organizing RBF network algorithm which optimizes the network structure to prevent over-fitting or underfitting of the data. In [11] we see that the number of basis functions increases gradually with each subsequent iteration. In [83] the structure of the network is varied by increasing or decreasing the number of basis functions based on adaptation laws developed using the lyapunov synthesis approach.

The hidden neurons in the neural network present another method with which to vary the structure. In [57], a variable structured NN based on fuzzy logic is implemented. The initial values are chosen by the user based on experience, then consequent structures are chosen based on the input-output mapping of the network. When any weight value is below a pre-set limit, that particular node in the network is deleted, this therefore means that the corresponding fuzzy rule is deleted from the fuzzy rule base and the structure of the network is then altered before the next iteration. This evaluative process is done with each iteration.

The research carried out in [95] proposes a novel mixed training algorithm which consists of error back propagation (EBP) and a variable structure system to optimize the way in which parameters are updated in neural networks. In the optimization of the number of neurons in the hidden layer, a new term based on the output of the hidden layer is added to the cost function in order to optimize the use of hidden units relative to the weights corresponding to each unit of the hidden layer. The advantages of this technique include guaranteed convergence, improved robustness and lower sensitivity to the initial weights of the neural network.

The network switch controller (NSC) which is a 3-layer feedforward neural network is also used to alter the structure of the neural network [80, 81]. In [80] the variable structure consists of a neural network with link switches (NNLS) and a network switch controller, the network switch controller is used to control the switch of the NNLS. The proposed structure would then be able to model different input patterns with variable network structures. A genetic algorithm (GA) is used for the tuning of the parameters of the neural networks. The research carried out in [81] consists of a neural network with node-to-node relationship (N^4R) and a network switch controller. In the N^4R, a modified neuron model which consists of two activation functions in the hidden layer and switches in its lines are introduced. The network switch controller controls the switches in the N^4R and therefore enables the proposed variable structure NN to be able to model different input patterns with a relatively low cost in terms of the computational cost and complexity of the system. Given the research conducted with regards to the variable structure neural networks, there are still rooms for improvement and this is the motivation for part of the research conducted in the thesis. The variable structured neural network is applied to deal with two relatively new applications (material classificaiton and epilepsy seizure phase classification). A novel VWNN structure is also proposed where the weights are varied over multiple layers compared to the weights being varied over a single layer in the literature.

2.1.6 Support Vector Machine

The support vector machine (SVM) [48, 50] is a kernel method that is used to map non-linear and inseparable data from an input space into a higher dimensional feature space where the data would then be linearly separable [143]. This is done with the aid of the separating hyperplane [152]. The benefit of the kernel method is that the use of kernel functions enables the user to save time and computational power as the computation in the higher dimensional feature space is no longer necessary [114]. The SVM algorithm aims to maximize the margin (the region separating the support vectors on either side of the hyperplane). This would result in an optimal classification accuracy of the hyperplane. Although the SVM sometimes suffers from high complexity and long computational times, it is shown to be very resistant to the problem of over-fitting the data. The SVM has a good generalization ability and also performs well in a high dimensional feature space. SVMs have been used for character recognition, text recognition and facial recognition [50].

2.1.6.1 Fuzzy Support Vector Machine

The fuzzy support vector machine (FSVM) is a hybrid classification method that was introduced to reduce the effect of outliers in the input on classification performance [138]. This is done by applying a degree of membership to each of the input datapoints which depicts the influence that the particular datapoint has on the overall output of the classifier. The objective of the FSVM is to reduce the membership of outliers in order to improve the overall result of the classifier, this constitutes the majority of research done in this field as is shown in the following paragraphs. The FSVM has been applied into numerous areas such as text categorization [20], image classification [74], image watermarking [75], evaluation of credit risk [43, 49, 56, 138] and hypoglycemia detection [98].

In [76] we see the FSVM being combined with the binary tree method. In the binary tree method, the dataset is split into a number of samples N which is defined by the user. The first sample is selected as the positive sample and the remaining samples are classed negative, a binary-valued SVM classifier is then trained to separate between both classes. This process is carried out iteratively through all the N samples. Another interesting application of the FSVM is in an image watermarking scheme [75] where the membership grade is calculated based on the texture strength of the image. The image is split into an 8*8 block and the texture features for each sub-block are used as the input vectors for the SVM, a sub-block with strong texture features is given a larger fuzzy membership than a sub-block with weaker texture. In [138] a multiclass FSVM is proposed where each sample of the input is treated as both a positive and negative class. An input is assigned with a low membership if it is detected as an outlier. The new FSVM also treats each input as an input of the opposite class, in this way, the FSVM makes full use of the data and is able to achieve a higher generalization ability.

A number of hybrid classifiers involving the FSVM have also been proposed in the literature, in [125] the FSVM is merged with the kNN technique to classify liver disorders. In this method, the membership function is not based solely on the distance between each datapoint and its class centre but also by a new measure known as the affinity among samples which can be defined with k-nearest neighbour distances. In [142] the FSVM is merged with fuzzy c-means clustering where the clustering algorithm is applied to each class of the training dataset. The clustering algorithm places datapoints of greater importance as the centre of the cluster. All the cluster centres now form the representation of other samples that are in close proximity to them and are used as the input to the FSVM classifier. This method is able to reduce the number of training data and also the time elapsed in finding a solution to the constrained optimization problem.

In [153] an improved algorithm is presented where sample points have different types of membership in different regions. The membership grade for a sample point near the class centre is determined based on the distance between the point and its class centre while the membership grade of a sample point that is far away from the class centre is determined by the proportion between the number of its congeneric points and the number of its heterogeneous points in its neighbourhood. The research conducted in [115] involves an iterative FSVM method where the membership grade is calculated based on the position of the training vector from the SVM decision surface. Training vectors that are closer to the decision surface have a larger impact than those further away. The FSVM is then trained again and a new membership grade is assigned based on the new SVM decision region. This process is repeated iteratively until the desired classification accuracy is obtained or a pre-defined number of iterations has been reached.

Although an extensive amount of research has been conducted in the FSVM field, there are still some particular areas lacking like that of the type-2 fuzzy logic based FSVM. In the literature there is a significantly sparse amount of published research in this field when compared to the type-1 fuzzy logic based FSVM. The type-2 FSVM however has a lot of advantages, a significant one being its ability to handle uncertainties in the input data. This is a very important ability as noisy datasets are an inevitability in majority of real-world applications. From the literature there is also a lack of application of the FSVM to deal with the epileptic seizure classification problem. This therefore forms a major motivation for the research conducted in this thesis as it addresses these deficiencies.

2.2 Background Theory

2.2.1 Artificial Neural Networks

A 3-layer feed-forward fully-connected neural network with $n_{n_{in}}$ inputs and $n_{n_{out}}$ outputs is shown in Fig. 2.1 where $w_{ji}^{(1)}$ denotes the weight between the j-th hidden node and the i-th input node; $w_{ji}^{(2)}$ denotes the weight between the j-th output node and the i-th hidden node, and $b_j^{(1)}$ and $b_j^{(2)}$ denote the weights of the biases in the j-th hidden and output nodes, respectively. It has been shown that a 3-layer feed-forward fully-connected neural network is a universal approximator [47] which can approximate a smooth and continuous non-linear function in a compact domain to an arbitrary level of accuracy.

A multiple-layer feed-forward fully-connected neural network with one input layer, n_l hidden layers and one output layer is briefly presented above. It takes $\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_{n_{in}}(t) \end{bmatrix}$ as the t^{th} input and produces $\mathbf{y}(t) = \begin{bmatrix} y_1(t) & y_2(t) & \cdots & y_{n_{out}}(t) \end{bmatrix}$ as the outputs where n_{in} denotes the number of input nodes in the input layer and n_{out} denotes the number of output nodes in the output layer.

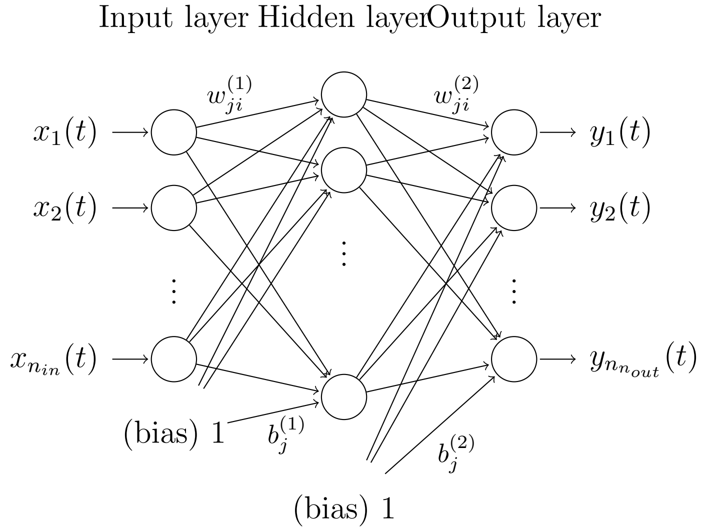

Figure 2.1: A three-layer feed-forward fully-connected neural network.

The output of the j-th node in the input layer is given as follows:

$$f_i^{(0)}(t) = x_i(t), i = 1, 2, \ldots, n_{in} \tag{2.1}$$

and the output of the j-th node in the n_l-th hidden layer is given as follows:

$$f_i^{(n_l)}(t) = tf_{n_l}\big(\sum_{j=1}^{n_{n_h}^{(n_l-1)}} w_{ij}^{(n_l)} f_j^{(n_l-1)}(t) - b_j^{(n_l)}\big), i = 1, 2, \ldots, n_h^{(n_l)}, \tag{2.2}$$

where $tf_{n_l}(\cdot)$ denotes the transfer function; $n_h^{(n_l)}$ denotes the number of hidden nodes, $b_i^{(n_l)}$ denotes the bias in the n_l-th hidden layer; and $w_{ij}^{(n_l)}$ denotes the weight between the j-th node in the $n_h^{(n_l-1)}$-th hidden layer and the i-th node in the $n_h^{(n_l)}$-th hidden layer.

The output of the neural network is given as follows:

$$y_i(t) = tf_{n_l+1}\big(\sum_{j=1}^{n_{n_h}^{(n_l)}} w_{ij}^{(n_l+1)} f_j^{(n_l)}(t) - b_j^{(n_l+1)}\big), i = 1, 2, \ldots, n_{out} \tag{2.3}$$

2.2.2 Support Vector Machines

The SVM theory is reviewed in this section, this provides the theoretical background to the development of IT2FSVM. The main objective of the SVM is to create a separating hyperplane such that the distance between the hyperplane and the nearest data point in each class is maximized.

Given a dataset S containing labelled training points

$$(y_1, x_1), \ldots, (y_N, x_N) \qquad i = 1, 2, \ldots, N \tag{2.4}$$

where vector x_i represents the training point, y_i represents the label and N represents the total number of samples. The vector x_i is assigned to either of two classes and is represented by the class label $y_i \in \{-1, 1\}$. The hyperplane is ideally placed in the middle of the margin between the two classes being separated. The data points that are in close proximity to the margin are the basis of its definition and are known as support vectors (SVs) [66]. In a non-linear function, searching for the optimum hyperplane in the input space is difficult. This is due to the fact that it there would be a high level of difficulty in defining a separating hyperplane to maximise the margin between points in a non-linear function when compared to a linear function. Hence, the input space is mapped onto a higher dimensional feature space where a maximum-margin hyperplane is defined. Even though the classifier is a hyperplane in the higher dimensional feature space, it may be non-linear in the original input space. Let $z = \varphi(x)$ represent the feature vector where x is an input vector and $\varphi(x)$ is a transformation function. The hyperplane can then be defined as

$$\omega \cdot z + b = 0 \tag{2.5}$$

where z is the feature space vector, ω is the weight vector and b is the scalar threshold (bias). The set S is linearly separable if there exists a combination of $\boldsymbol{\omega}$ and b that satisfies the following inequalities for all elements of the set S.

$$\begin{cases} \omega \cdot z_i + b \geq 1, & \text{if}\, y_i = 1 \\ \omega \cdot z_i + b \leq -1, & \text{if}\, y_i = -1, \qquad i = 1, 2, \ldots, N \end{cases} \tag{2.6}$$

where $z_i = \varphi(x_i)$.

As the set S is not linearly separable for all of its elements, a leeway for some classification violations must be allowed in order to accommodate the elements of the set that are not linearly separable. This deficiency can be resolved by introducing non-negative slack variables $\xi_i \geq 0$ for the samples x_i which do not satisfy (2.6). Hence, (2.6) is then modified to

$$\begin{cases} \omega \cdot z_i + b \geq 1 - \xi_i, & \text{if}\, y_i = 1 \\ \omega \cdot z_i + b \leq -1 - \xi_i, & \text{if}\, y_i = -1, \qquad i = 1, 2, \ldots, N \end{cases} \tag{2.7}$$

The optimal hyperplane can be obtained as a solution to the constrained optimization problem

$$min \qquad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\xi_i \tag{2.8}$$

subject to

$$y_i(\omega \cdot z_i + b) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, N \tag{2.9}$$

$$\xi_i \geq 0 \qquad i = 1, 2, \ldots, N \tag{2.10}$$

where (2.8) is the convex cost function, (2.9) and (2.10) are the constraints, $\| \cdot \|$ denotes the l^2 norm (i.e. Euclidean norm), and C is known as the regularization constant which is the only free parameter in the SVM formulation and can be tuned to find a balance between margin maximization and classification violation. The optimal hyperplane can be found by constructing a Lagrangian multiplier and obtaining the dual formation:

$$min \qquad Q(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j z_i \cdot z_j - \sum_{i=1}^{N} \alpha_i \tag{2.11}$$

subject to

$$\sum_{i=1}^{N} y_i \alpha_i = 0, \qquad 0 \leq \alpha_i \leq C, \qquad i = 1, 2, \ldots, N \tag{2.12}$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)$ represents the vector of the nonnegative langrange multipliers which satisfy the constraints in (2.8).

Karush-Kuhn-Tucker theorem[139] is important to the development of the SVM. The theorem states that the solution α_i to (2.12) satisfies the following conditions:

$$\alpha_i(y_i(\omega \cdot z_i + b) - 1 + \xi_i) = 0, \qquad i = 1, 2, \ldots, N \tag{2.13}$$

$$(C - \alpha_i)\xi_i = 0, \qquad i = 1, 2, \ldots, N \tag{2.14}$$

The equalities (2.13) and (2.14) suggest that it is only the nonzero values α_i in (2.11) that satisfy the constraints in (2.9). The values of x_i that corresponds with the solution α_i are known as support vectors. The instance is correctly classified when x_i corresponds with $\alpha_i = 0$ and is a significant distance away from the decision margin.

For the construction of the optimal hyperplane $\omega \cdot z + b$, we would require that

$$\omega = \sum_{i=1}^{N} \alpha_i y_i z_i \tag{2.15}$$

and the scalar bias b should be determined via the Karush-Kuhn-Tucker conditions in (2.13).

The decision function can then be derived from (2.6) and (2.15) as

$$f(x) = \operatorname{sgn}(\omega \cdot z + b) = sgn\big(\sum_{i=1}^{N} \alpha_i y_i z_i \cdot z + b\big) \tag{2.16}$$

where $\operatorname{sgn}(\cdot)$ represents the sign function which extracts the sign (positive or negative) of a real number. As we have no knowledge of the higher dimensional feature space $\varphi(\cdot)$, carrying out the computation in (2.11) and (2.16) would be rendered impossible due to its complicated nature. An advantageous characteristic of the SVM is that it is not necessary to determine $\varphi(\cdot)$. The problem is alleviated with the aid of a kernel function which has the ability to compute the dot product of the data points in the feature space of z. It is however obligatory for these functions to satisfy Mercer's theorem [78] before they can be used for computing the dot product[79].

$$z_i \cdot z_j = \varphi(x_i) \cdot \varphi(x_j) = K(x_i, x_j) \tag{2.17}$$

where $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ is the kernel function which is used for the mapping onto a higher dimensional feature space. The kernel functions can be linear or nonlinear. The nonlinear separating hyperplane can be determined by solving the following equation

$$min \qquad Q(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i \tag{2.18}$$

subject to

$$\sum_{i=1}^{N} y_i \alpha_i = 0, 0 \leq \alpha_i \leq C, \qquad i = 1, 2, \ldots, N. \tag{2.19}$$

The decision function can then be described as follows:

$$f(x) = \text{sgn}(\omega \cdot z + b) = sgn\Big(\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b \Big) \tag{2.20}$$

2.2.3 Interval Type-2 Fuzzy Inference System (IT2FIS)

Fuzzy inference systems are mainly used to represent the relationship between the input and output variables in systems. Fuzzy inference systems are governed by selecting IF-THEN rules which utilize linguistic labels for the expression of rules and facts. An IT2FIS is a fuzzy logic system where the uncertainty of the membership functions are incorporated into fuzzy set theory. In the circumstance where no uncertainty exists, a type-2 fuzzy set would reduce to a type-1 fuzzy set, this is identical to the concept of probability reducing to the determinism when the unpredictability is eradicated [91]. In order to distinguish between a type-1 and type-2 fuzzy set, a tilde symbol is placed above the symbol for the fuzzy set, in this case, A represents a type-1 fuzzy set and $\tilde{A}$ represents a type-2 fuzzy set [96].

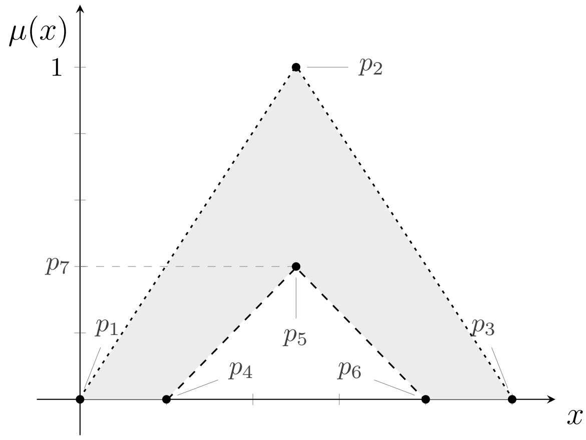

Figure 2.2: An example of IT2 membership functions. Dashed line: lower membership function. Dotted line: Upper membership function. Gray area: footprint of uncertainty.

An example triangular IT2FIS membership function is shown in Fig. 2.2. The shape of the membership function is a triangle, with the dashed lines representing the lower membership function LMF and the dotted line representing the upper membership function UMF. The membership function can either be predefined by the users or designed with the aid of optimization methods such as the genetic algorithm (GA). The membership function for each input is represented by seven points (p_1 to p_7) which can be optimised by the GA. Unlike in the type-1 case where the membership grade is a crisp value, the membership grade in an IT2FIS is an interval. The IT2FIS is then bounded at the two extremes of this interval to give us the LMF and UMF which are both type-1 fuzzy sets. The area between the UMF and LMF is known as the footprint of uncertainty (FOU) which is shown as the gray area in Fig. 2.2.

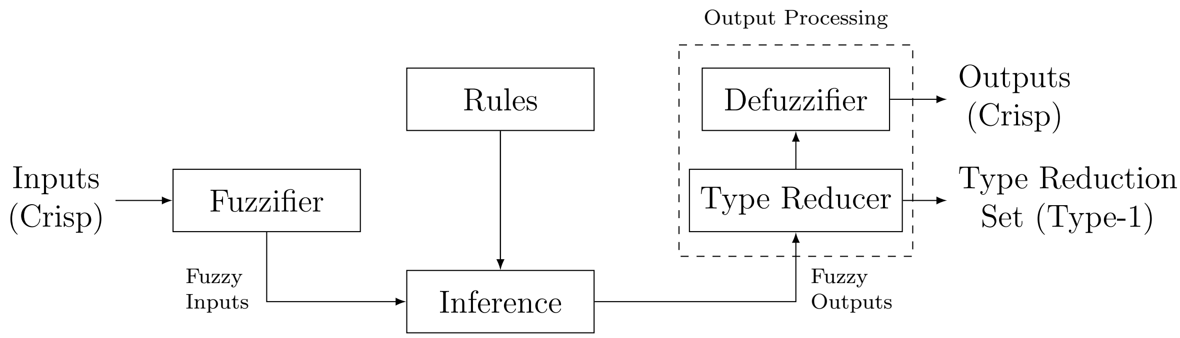

Figure 2.3: Block diagram showing the IT2FIS [9].

Type-2 fuzzy sets are more prevalent than type-1 fuzzy sets in rule-based fuzzy logic systems as they have a higher level of non-linearity and therefore type-2 fuzzy sets have the ability to model uncertainties better than the type-1 fuzzy sets with less number of rules[126]. The structure of the IT2FIS detailing the input-output relationship is shown in Fig. 2.3. The IT2FIS consists of 5 major components [140]: fuzzifier, fuzzy rules, inference engine, type-reducer and defuzzifier. The crisp input is first transformed into fuzzy sets in the fuzzifier block as the rule base is activated by fuzzy sets and not numbers. In the fuzzification stage, when the measurements are perfect the input is modelled as a crisp data set, when the measurements are noisy but stationary it is modelled as an interval type-2 fuzzy set. After the input is fuzzified, the fuzzy input set is then mapped onto the fuzzy output set with the aid of the inference block. This is achieved by quantifying each rule using fuzzy set theory and then using the mathematics behind fuzzy set theory to obtain an output for each rule. The output of the fuzzy inference block would then contain one or more fuzzy output sets. The fuzzy output sets are then converted into a crisp output with the aid of the output processing unit. In an IT2FIS the output processing unit consists of two blocks: the type-reducer and the defuzzifier blocks. In the first step, the IT2 fuzzy output set is reduced to an interval-valued type-1 fuzzy set through type-reduction.

Given an IT2FIS with n inputs $x_i \in X_i, \ldots, x_n \in X_n$ to give a singular output $y \in Y$. The rule base for this IT2FIS consists of K IT2 fuzzy rules expressed in the following form [132]:

$$R^k : \text{If } x_1 \text{ is } \tilde{F}_1^k \text{ and } \cdots \text{ and } x_n \text{ is } \tilde{F}_n^k \text{ THEN } y \text{ is } \tilde{G}^k \tag{2.21}$$

where $k = 1, \ldots, K$, $\tilde{F}_n^k$ and $\tilde{G}^k$ represent type-2 fuzzy sets.

The rules are responsible for the mapping of an input domain X to an output domain Y. Experimentation has shown that the general T2FIS model has high computational costs and complexity. This has resulted in the development of the IT2FIS which makes the computation simplified. The membership grades for interval fuzzy sets can be portrayed by their lower and upper membership grades of the FOU. The output of the firing strength for an IT2FIS ω_i is represented by a lower and upper bound i.e., $\omega_i = [\underline{\omega}_i, \overline{\omega}_i]$. The defuzzified output is obtained by type reduction which is implemented using the KM algorithm [140] given in Section 2.2.4 :

2.2.4 KM Algorithm

The first step of defuzzification is type reduction, where a type-2 fuzzy set is reduced to a type-1 fuzzy set. The KM algorithm which was developed by Karnik and Mendel [140] is an example of such a method. The KM algorithm is iterative and has fast convergence rates, hence its suitability for the research conducted in this thesis. The iterative procedure produces an upper and lower bound of the output. The second step of output processing occurs after type-reduction. In the case of the KM algorithm being used as a type-reducer, the type-reduced set is confined to a finite interval of numbers, the deffuzifier then obtains the defuzzified value (which is a crisp output) by calculating the average of the upper and lower bounds of this interval. A detailed description of the KM algorithm is shown below in Section 2.2.4.1 and Section 2.2.4.2.

2.2.4.1 Lower Bound

1. Determine the lower bound of the output $\underline{x}_i (i = 1, \ldots, n)$ in ascending order and then assign the same labels to them such that $\underline{x}_1 \leq \underline{x}_2 \leq \cdots \leq \underline{x}_n$.

2. Match the weights ω_i with the corresponding $\underline{x}_i$ and reassign the labels to match with the new $\underline{x}_i$ which are now in ascending order.

3. Initialize ω_i, i.e.,

$$\omega_i = \frac{\underline{\omega}_i + \overline{\omega}_i}{2} \text{ where } i = 1, \ldots, n \tag{2.22}$$

then calculate

$$y = \frac{\sum_{i=1}^{n} \underline{x}_i \omega_i}{\sum_{i=1}^{n} \omega_i} \tag{2.23}$$

4. Determine the pivot point p where $(1 \leq p \leq N-1)$ such that

$$\underline{x}_p \leq y \leq \underline{x}_{p+1} \tag{2.24}$$

5. Assign the firing strength as

$$\begin{cases} \overline{\omega}_i, & i \leq p \\ \underline{\omega}_i, & i > p \end{cases} \tag{2.25}$$

then calculate

$$y' = \frac{\sum_{i=1}^{n} \underline{x}_i \omega_i}{\sum_{i=1}^{n} \omega_i} \tag{2.26}$$

6. Check if $y' = y$; If yes, stop and set $\underline{y} = y$, if no, go to step 7

7. Set $y = y'$ and go to step 3

2.2.4.2 Upper Bound

1. Define the upper bound of the output $\overline{x}_i (i = 1, \ldots, n)$ in ascending order and then assign the same labels to them such that $\overline{x}_1 \leq \overline{x}_2 \leq \ldots \leq \overline{x}_n$.

2. Match the weights ω_i with the corresponding $\overline{x}_i$ and reassign the labels to match with the new $\overline{x}_i$ which are now in ascending order.

3. Initialise ω_i i.e

$$\omega_i = \frac{\underline{\omega}_i + \overline{\omega}_i}{2} \text{ where } i = 1, \ldots, n \tag{2.27}$$

then calculate

$$y = \frac{\sum_{i=1}^{n} \overline{x}_i \omega_i}{\sum_{i=1}^{n} \omega_i} \tag{2.28}$$

4. Determine the pivot point p where $(1 \leq p \leq N-1)$ such that

$$\overline{x}_p \leq y \leq \overline{x}_{p+1} \tag{2.29}$$

5. Assign the firing strength as

$$\begin{cases} \underline{\omega}_i, & i \leq p \\ \overline{\omega}_i, & i > p \end{cases} \tag{2.30}$$

then calculate

$$y' = \frac{\sum_{i=1}^{n} \overline{x}_i \omega_i}{\sum_{i=1}^{n} \omega_i} \tag{2.31}$$

6. Check if $y' = y$; If yes, stop and set $\overline{y} = y$, if no, go to step 7

7. Set $y = y'$ and go to step 3

The defuzzified output of the IT2FIS is given as:

$$y = \frac{\overline{y} + \underline{y}}{2} \tag{2.32}$$

Chapter 3

Neural Network Architectures for Surface Classification - Neural-Network-Based Classifiers and Their Applications

3.1 Introduction

In this chapter, we consider a classification problem in material surface classification of an unknown object using a contact sensing fingertip, which demonstrates a wide range of potential domestic and industrial applications, such as on robot-assisted surgery [5, 64, 100, 134], blind grasping application [22, 101], pose classification [85], prosthetic limbs [27], quality assurance [54], shape extraction and industrial inspection [14, 128], and brain-machine-brain interface [99]. The properties of the object surface which are important for the aid of classification are the frictional coefficients, texture, compliance and roughness. The data is obtained through an active surface exploration method [116, 117] with the aid of contact-sensing fingertip which can accurately identify the normal and frictional force of the object. During the experiments, the contact sensing fingertip slides along the object with short strokes whilst increasing/decreasing the velocity as is appropriate. The properties of the vibrations caused by this action are then used as the input data. A feature vector is extracted from the raw data to reduce the number of data points used for the classification procedure. It is of utmost importance that the contact sensing fingertip is able to differentiate between the objects and that is the basis of emphasis and importance for the research being conducted in this chapter.

In view of the superior learning and generalization capability of the neural networks, we are motivated to implement classifiers using neural networks to deal with the material classification problem [69, 70, 71]. In this study, the characteristics of the neural networks are considered for the implementation of neural-network-based classifiers, demonstrating different levels of flexibility, scalability and complexity. Six neural-networked-based classifiers, namely one-against-all, weighted one-against-all, binary coded, parallel structured, weighted parallel structured, tree-structured, are introduced for classification of materials touched by the robot finger. In order to make a comparison, two traditional classification methods, namely k-nearest neighbor classifier and the naive Bayes classifier, are considered. Their classification performance is investigated thoroughly using the dataset collected from experiments. To investigate the robustness property of the classifiers, Gaussian white noise is added to the test dataset and the classification performance is evaluated. By investigating the classification performance of the introduced classifiers, the most suitable classifier for the material surface classification problem can be recommended.

This chapter is arranged as follows: After the introduction, Section 3.2 presents the six neural network based classifiers and comments on their flexibility, scalability and complexity. The robustness of all the classifiers are also included. The material classification problem is discussed in Section 3.3. Section 3.4 presents the results produced from the simulations under both the original testing data case and noisy data case. The results are discussed in Section 3.5 and a conclusion is then drawn in Section 3.6.

3.2 NN-Based Classifiers

In this section, six NN-based classifiers namely one-against-all, weighted one-against-all, binary coded, parallel-structured, weighted parallel-structured and tree-structured, are introduced to classify the feature patterns. In the following, the input pattern is denoted as $\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \ldots \quad x_{n_{in}}(t)]$, which is considered as the feature vector of an object to be recognised. The purpose of these classifiers is to group the feature patterns into M classes through supervised learning. The six architectures that have been proposed we chosen from an abundance of existing methods in the literature. However these six were chosen due to their suitability and practicality in being a viable approach to deal with the material classification problem. The structure of the proposed architectures were then adjusted in order to fit this problem space and this represents a key contribution of the research that has been conducted in this chapter.

3.2.1 One-Against-All Classifier

A one-against-all classifier is shown in Fig. 3.1, which can be considered as a multiple-input-single-output fully-connected feed-forward NN. It receives the feature pattern $\mathbf{x}(t)$ as input and produces a single value $y(t)$ as output. The target output $y^d(t)$ is set to be i when the input pattern $\mathbf{x}(t)$ belongs to class i. In other words, the one-against-all classifier is trained such that the output $y(t)$ is as close as possible to $y^d(t)$ according to the class of the feature pattern $\mathbf{x}(t)$.

During the operation, the feature pattern is classified as of class j which is obtained by

$$j = \arg\min_i \{|y(t) - i| \mid i \in \{1, 2, \ldots, M\}\}, \tag{3.1}$$

where $|\cdot|$ is the absolute value operator. If the set j has more than one element, the first element is considered as the recognised class label.

The one-against-all classifier has a simple structure. However, it is less flexible and retraining is required when additional classes are introduced. Also, when the number of classes increases, the training time increases accordingly. For a large-scale classification problem (for example, with large dataset, large number of classes and/or high dimensional input features), the number of hidden nodes and/or layers have to be increased to achieve an acceptable classification accuracy.

3.2.2 Weighted One-Against-All Classifier

A weighted one-against-all classifier is shown in Fig. 3.2, which can be considered as a multiple-input-multiple-output fully-connected feed-forward NN. It receives a feature pattern $\mathbf{x}(t)$ as input and produces a vector $\mathbf{y}(t) = [y_1(t) \quad y_2(t) \quad \ldots \quad y_{n_{out}}(t)]$

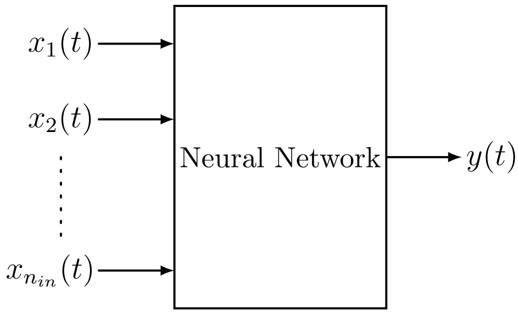

Figure 3.1: NN-based one-against-all classifier.

as output where n_{out} is a non-zero positive integer pre-determined by designers. The target output vector $\mathbf{y}^d(t) = \begin{bmatrix} y_1^d(t) & y_2^d(t) & \dots & y_{n_{out}}^d(t) \end{bmatrix}$ is set to be $\mathbf{w}_i = [w_{i1} \quad w_{i2} \quad \dots \quad w_{in_{out}}]$, $i = 1, 2, \dots, M$, which is a predefined constant vector to be determined, when the input pattern $\mathbf{x}(t)$ belongs to class i. During the operation, the input pattern is classified as of class j which is obtained by

$$j = \arg\min_i \{\|\mathbf{y}(t) - \mathbf{w}_i\| \mid i \in \{1, 2, \dots, M\}\}, \tag{3.2}$$

where $\|\cdot\|$ denotes the l^2 norm (i.e. Euclidean norm). If the set j has more than one element, the first element is considered as the recognised class label.

Compared with the one-against-all classifier, it offers a relatively higher flexibility to assign the target output, which could improve the classification accuracy by examining more than one output. As weighted one-against-all classifier is based on the one-against-all classifier, it inherits the same limitations in terms of flexibility, scalability and complexity as in one-against-all classifier discussed in Section 3.2.1.

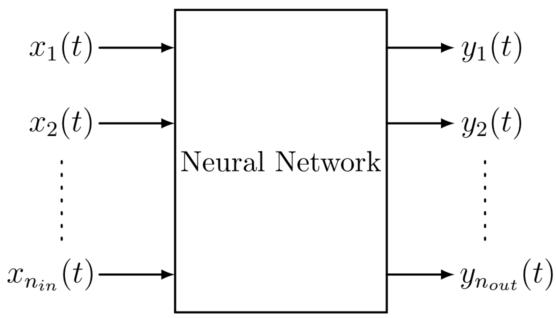

Figure 3.2: NN-based weighted one-against-all or binary-coded classifier.

3.2.3 Binary-Coded Classifier

A binary-coded classifier is shown in Fig. 3.2, which can be considered as a multiple-input-multiple-output fully-connected feed-forward NN. It receives a feature pattern $\mathbf{x}(t)$ as input and produces a vector $\mathbf{y}(t) = [y_1(t) \quad y_2(t) \quad \dots \quad y_{n_{out}}(t)]$ as output where $n_{out} = \left\lceil \frac{\log M}{\log 2} \right\rceil$, $\lceil \cdot \rceil$ denotes the ceiling operator rounding up the argument to the nearest integer. To reduce the number of outputs of the NN, binary string is employed to represent the class of the input patterns. Class i, $i = 1, 2, \dots, M$, is represented by an n_{out}-bit binary string. For example, assuming that $M = 18$, a 5-bit binary string is employed to represent the class of input patterns; class 1 is represented by '00001', class 2 is represented by '00010' and so on. The target output vector $\mathbf{y}^d(t) = \left[y_1^d(t) \quad y_2^d(t) \quad \dots \quad y_{n_{out}}^d(t)\right]$ is set to be $\mathbf{w}_i = [w_{i1} \quad w_{i2} \quad \dots \quad w_{in_{out}}]$, $i = 1, 2, \dots, M$, which is the binary representation of i.

The binary-coded classifier is a subset of the weighted one-against-all classifier. When the weight $\mathbf{w}_i$ of the weighted one-against-all classifier is chosen to be a binary string, the classifier is configured as the binary-coded classifier. During the operation, the input pattern is recognised as of class j based on (3.2).

3.2.4 Parallel-Structured Classifier

A parallel-structured classifier is shown in Fig. 3.3, which consists of M n_{in}-input-n_{in}-output fully-connected feed-forward NNs. Fig. 3.3 shows that the purpose of the i^{th} NN is to recognise the input patterns of class i. To realise this purpose, the training objective is that the output of the NN corresponding to class i is the same as the input patterns of class i, i.e., the target output vector $\mathbf{y}^d(t)$ is set to be $\mathbf{x}(t)$ such that the characteristic of input patterns of class i can be learnt. Consequently, it is expected that the difference between the input and output vector of the i^{th} NN would be very small if the input patterns are of class i but relatively larger if the input pattens are not of class i. The class determiner in Fig. 3.3 will determine the input patten to be of class i if the i^{th} NN produces the least input-output difference. During the operation, the feature pattern is classified as class j which is obtained by

$$j = \arg\min_{i}\{\|\mathbf{y}_i(t) - \mathbf{x}(t)\| \mid i \in \{1, 2, \ldots, M\}\}. \tag{3.3}$$

If the set j has more than one element, the first element is considered as the recognised class label.

The i^{th} NN is trained with the feature patterns of class i implying that the complexity of the NN is lower compared with those in one-against-all, weighted one-against-all and binary-coded classifier that feature patterns of all classes are used for training the NN. Moreover, it is more flexible to add extra classes and retraining of all existing NNs is not necessary. It is thus more suitable to handle large-scale classification problem.

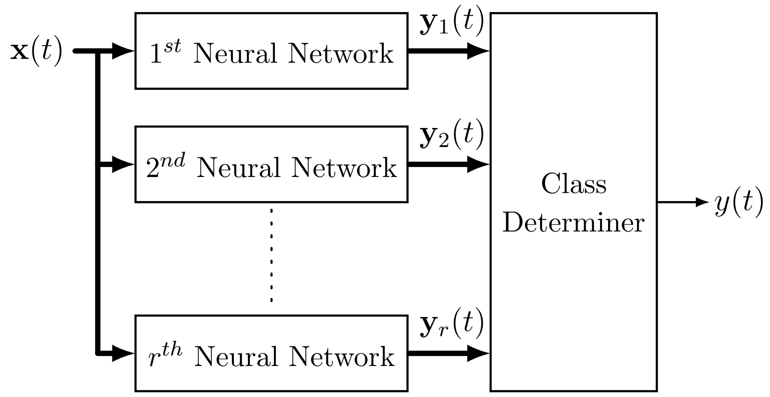

Figure 3.3: NN-based parallel-structured classifier.

3.2.5 Weighted Parallel-Structured Classifier

A weighted parallel-structured classifier is a variant of parallel-structured classifier, which consists of $\lceil \frac{M}{G} \rceil$ n_{in}-input-n_{in}-output fully-connected feed-forward NNs. Each NN in the parallel-structured classifier is able to learn the characteristic of one single class of input patterns and the classification is realised by looking into the least input-output difference. The weighted parallel-structured classifier allows each NN to learn the characteristic of more than one class of feature patterns such that each NN can classify more than one class. It reduces the number of NNs to implement the weighted parallel-structured classifier.

Let $G \leq M$ be the number of classes recognised by each NN. The i^{th} NN is trained such that the target output vector $\mathbf{y}^d(t)$ is set to be $\mathbf{W}_k\mathbf{x}(t)$ where $k = (i-1)\lceil \frac{M}{G} \rceil + 1,\ (i-1)\lceil \frac{M}{G} \rceil + 2,\ \ldots,\ (i-1)\lceil \frac{M}{G} \rceil + G,\ i = 1,\ 2,\ \ldots,\ \lceil \frac{M}{G} \rceil$, when the feature pattern $\mathbf{x}(t)$ belongs to class k; $\mathbf{W}_k = \text{diag}\{w_{k1}, w_{k2}, \ldots, w_{kn_{in}}\}$ is a constant matrix determined by the designers. Consequently, when the input pattern $\mathbf{x}(t)$ is input to the i^{th} NN, the l^2 norm of the difference between the weighted input and output, i.e., $\|\mathbf{y}_i(t) - \mathbf{W}_k\mathbf{x}(t)\|$ should be very small when $\mathbf{x}(t)$

belongs to class k, otherwise, a relatively larger l^2 norm of the difference should be obtained. The class determiner will determine the class of the input pattern based on the least l^2 norm of the difference.

During the operation, the feature pattern is classified as of class j which is obtained by

$$\begin{aligned} j = \arg\min_k \{ &\|\mathbf{y}_i(t) - \mathbf{W}_k \mathbf{x}(t)\| \mid i \in \{1, 2, \ldots, \lceil \frac{M}{G} \rceil\}; \\ &k \in \{(i-1)\lceil \frac{M}{G} \rceil + 1, (i-1)\lceil \frac{M}{G} \rceil + 2, \ldots, (i-1)\lceil \frac{M}{G} \rceil + G\}\}. \end{aligned} \tag{3.4}$$

If the set j has more than one element, the first element is considered as the recognised class label.

3.2.6 Tree-Structured Classifier

A tree-structured classifier is shown in Fig. 3.4, which consists of a single group classifier and $\lceil \frac{M}{G} \rceil$ sub-classifiers making a total of $1 + \lceil \frac{M}{G} \rceil$ NNs. We firstly divide the total number of classes into $\lceil \frac{M}{G} \rceil$ groups such that each group has G sub-classes. The group classifier is an n_{in}-input-$\lceil \frac{M}{G} \rceil$-output NN. The group classifier indicates which group the input pattern belongs to and then select the corresponding sub-classifier to perform classification. During the training, the target output $z_k^d(t)$ for output $z_k(t)$, $k = 1, 2, \ldots, \lceil \frac{M}{G} \rceil$, is set to be 1 if the input pattern belongs to group k, otherwise, 0. When output $z^k(t)$ of the group classifier is closer to 1, which suggests that the input pattern belongs to group k, the k^{th} sub-classifier is selected to determine which sub-class the input pattern belongs to in this group.

During the operation, the feature pattern is classified as of group j which is obtained by

$$j = \arg\min_{k}\{|z_k(t) - 1| \mid k \in \{1, 2, \ldots, \lceil \frac{M}{G} \rceil\}\}. \tag{3.5}$$

If the set j has more than one element, the first element is considered as the recognised class label.

After the input pattern is recognised as of group j, the j^{th} sub-classifier indicates which sub-class the input pattern belongs to. The sub-classifier is an n_{in}-input-G-output NN. The l^{th} output of sub-classifier being 1 is to indicate the input pattern belongs to sub-class l in group j so that the actual class of the input pattern is $(j-1)G + l$. Based on this mechanism, the target output $y_k^d(t)$ for output $y_k(t)$, $k = 1$, 2, ..., G, is set to be 1 if the input pattern belongs to sub-class k, otherwise, 0.

During the operation, the input pattern is classified as of sub-class l which is obtained by

$$l = \arg\min_{k}\{|y_k(t) - 1| \mid k \in \{1, 2, \ldots, G\}\}. \tag{3.6}$$

If the set l has more than one element, the first element is considered as the recognised class label.

The tree-structured classifier provides flexibility to add extra classes without retraining the sub-classifiers, however, the group classifier has to be retrained. Furthermore, the number of levels can be increased to deal with large-scale classification problems. As the classification error propagates to the lower levels, the classification performance of the upper-level classifiers, i.e., the group classifier, plays an important role to the overall classification performance of the tree-structured classifier. Unlike other classifiers introduced above, the processing time for classification is relatively longer as the lower-level classifiers cannot start to work until result has been received from the upper levels. As the sub-classifiers only need to deal with sub-classes, the complexity of NN is relatively lower compared with the classifiers with a single NN.

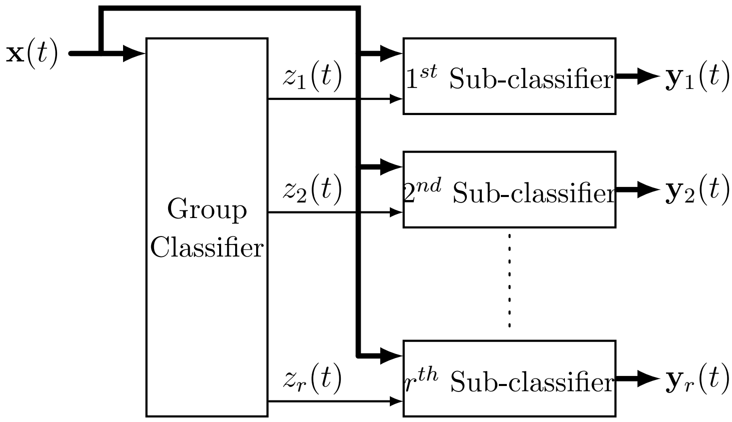

Figure 3.4: NN-based tree-structured classifier.

The properties of the NN-classifiers are summarised in Table 3.1, which compares the number of NNs used, number of outputs of NNs, flexibility adding extra classes, scalability in handling large-scale classification problems and complexity of NNs used in the classifiers.

3.3 Material Classification

In this section we consider a classification problem of the material surface classification of an unknown object using a contact sensing fingertip. Material surface classification has a wide range of potential domestic and industrial applications,

Classifier	#NNs	#outputs	Flexibility	Scalability	Complexity
1	1	1	Low	Low	High
2	1	n_{out}	Low	Low	High
3	1	$\left\lceil \frac{\log M}{\log 2} \right\rceil$	Low	Low	High
4	M	n_{out}	High	High	Low
5	$\left\lceil \frac{M}{G} \right\rceil$	n_{out}	High	High	Medium
6	$1 + \left\lceil \frac{M}{G} \right\rceil$	$\left\lceil \frac{\log M}{\log 2} \right\rceil$ or $\left\lceil \frac{M}{G} \right\rceil$	Medium	Medium	Medium

Table 3.1: Comparison of various NN-based classifiers. Classifier 1: one-against-all classifier, classifier 2: weighted one-againsy-all classifier, classifier 3: binary-coded classifier, classifier 4: parallel-structured classifier, classifier 5: parallel-structured classifier, classifier 6: tree-structured classifier.

examples include robot-assisted surgery [5, 64, 100, 134], blind grasping application [22, 101], pose classification [85], prosthetic limbs [27], quality assurance [54], shape extraction and industrial inspection [14, 128] and brain-machine-brain interface [99].

A classifier is implemented to classify the 18 materials listed in Table 3.2 using data collected from a robotic testing platform shown in Fig. 3.5, which includes a robot arm Mitsubishi RV-6SL, a 6-axis force/torque sensor ATI Nano17 (resolution = 0.003 N, sampling rate = 100 Hz) and a hemispherical plastic fingertip. During experiments, the fingertip which is rigidly attached to the robot arm was kept perpendicular to the material surface all the time. It was then commanded to slide on a selected object surface, keeping the normal force around 2 N. To obtain the dynamic relationship of friction and velocity, within one stroke, the sliding velocity was increased from zero to 15 mm/s with a constant acceleration rate of 3mm/s2. Each time the fingertip slides along a material surface, 100 numerical values (raw data of frictional force) reflecting the material characteristics are collected. The same experiment was repeated for 60 times for each of the 18 materials. In total, 60 sets of data (each set contains 100 numerical values) for each material were collected. Further detailed description of the experiment setup and data collection can be found in [68, 84].

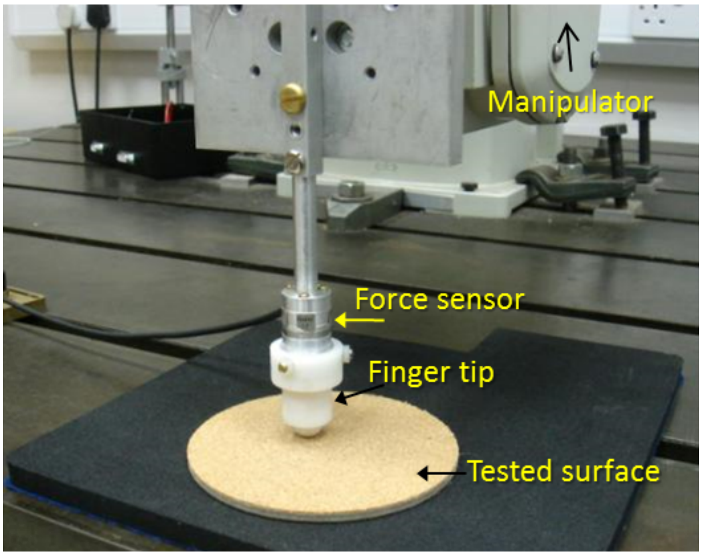

Figure 3.5: The test platform.

3.3.0.1 Feature Extraction

In these experiments, the raw data of 100 points (denoted as p_1 to p_{100}) will first be reduced to feature vectors of 3, 4 and 5. As a result, the raw data of each pattern (100 numerical values)is represented by 3, 4 or 5 features, which significantly reduces the dimensions of the input features, implying reduced computational demand and implementation complexity. The raw data of 100 numerical values

Class label	Material
1	Un-laminated wood
2	Fine polished aluminium
3	Unpolished aluminium
4	Polished brass
5	Ceramic plate
6	Cloth liner
7	Glass
8	Artificial leather
9	Mouse pad (liner surface)
10	A4 paper
11	Laminated book cover
12	Plastic PC mouse
13	Plastic CD cover
14	Polymer composite (smooth surface)
15	Kitchen sponge
16	Stainless steel knife
17	Rubber tape
18	Un-laminated paper package

Table 3.2: 18 Materials used in the experiment.

of each pattern is first divided into P portions where $P = 4$ such that $\mathbf{p}_1 = [p_1 \quad p_2 \quad \dots \quad p_{25}]$, $\mathbf{p}_2 = [p_{26} \quad p_{27} \quad \dots \quad p_{50}]$, $\mathbf{p}_3 = [p_{51} \quad p_{52} \quad \dots \quad p_{75}]$ and $\mathbf{p}_4 = [p_{76} \quad p_{77} \quad \dots \quad p_{100}]$. Define

$$f_1(\mathbf{z}) = \frac{1}{S}\sum_{i=1}^{S} z_i, \tag{3.7}$$

$$f_2(\mathbf{p}) = \sum_{i=1}^{4} |f_1(\mathbf{p}_{i+1}) - f_1(\mathbf{p}_i)|, \tag{3.8}$$

$$f_3(\mathbf{z}) = \frac{1}{S-1}\sum_{i=1}^{S}(z_i - f_1(\mathbf{z}))^2, \tag{3.9}$$

where $\mathbf{z} = [z_1 \quad z_2 \quad \ldots \quad z_S]$ and S is an integer representing the number of elements in $\mathbf{z}$.

Based on the functions in (3.7) to (3.9), we define the feature vectors of 3 to 5 points as follows:

Feature vector with 3 points:

$$\mathbf{x} = \left[\begin{array}{ccc} \sum_{i=1}^{4} f_1(\mathbf{p}_i) & 50 f_2(\mathbf{p}_i) & 50\sum_{i=1}^{4} f_3(\mathbf{p}_i) \end{array}\right]. \tag{3.10}$$

Feature vector with 4 points:

$$\mathbf{x} = \left[\begin{array}{cccc} \sum_{i=1}^{4} f_1(\mathbf{p}_i) & 50 f_2(\mathbf{p}_i) & 50\sum_{i=1}^{4} f_3(\mathbf{p}_i) & 20\sum_{i=1}^{4} \sqrt{f_3(\mathbf{p}_i)} \end{array}\right]. \tag{3.11}$$

Feature vector with 5 points:

$$\mathbf{x} = \left[\begin{array}{ccc} \sum_{i=1}^{4} f_1(\mathbf{p}_i) & 50|f_1(\mathbf{p}_2) - f_1(\mathbf{p}_1)| & 50|f_1(\mathbf{p}_3) - f_1(\mathbf{p}_2)| \\ 50|f_1(\mathbf{p}_4) - f_1(\mathbf{p}_3)| & 50\sum_{i=1}^{4} f_3(\mathbf{p}_i) & \end{array}\right]. \tag{3.12}$$

It can be seen from (3.7) to (3.9) that $f_1(\mathbf{z})$ is the mean of $\mathbf{z}$, $f_2(\mathbf{z})$ is the sum of the difference of the mean of the consecutive portions of raw data, $f_3(\mathbf{z})$ is the variance of $\mathbf{z}$. The classification performance of the introduced NN-based classifiers is investigated in its application to this problem in the following section.

3.4 Experimental Results

3.4.1 NN-based Classification

The 6 NN-based classifiers are employed to recognise the 18 materials using the feature vectors of 3, 4, and 5 points. The introduced classifiers were implemented on Matlab. The Levenberg-Marquardt back-propagation [118] is used to train the classifiers by minimizing the mean square error.

In this experiment, recalling that 60 sets of raw data are being collected for each material, 40 of them are to be used for the training of NNs and 20 of them are used for testing. Various transfer functions and different number of hidden nodes and hidden layers have been tried in this study. In the following, only the appropriate configurations (number of hidden nodes, transfer functions, etc.) which can achieve acceptable classification accuracy are reported. The linear transfer function is used in the output layer of all classifiers. For comparison purposes, the traditional kNN classifier and naive Bayes classifier are employed for the classification problem. To investigate how the noise influences the classification performance of the classifiers, which is inevitable in real world, the test dataset contaminated by Gaussian white noise with variance of 0.005 was considered. It should be noted that the simulations for all classifiers tested with noisy test dataset are conducted 10 times for fair comparison, as different solutions can be obtained by the Levenberg-Marquardt back-propagation algorithm with different initial guesses. Statistical information of the tests including the average classification accuracy for individual class, maximum and minimum classification accuracy and standard deviation of the 10 tests are reported.

In the following, the classification performance of the 6 NN-based classifiers, kNN classifier and naive Bayes classifier for the classification problem subject to original and noisy datasets is reported. The average classification accuracies for the training and testing datasets are summarised in Table 3.3. Referring to this table, the "Average" column reports the average training/testing classification accuracy for 18 materials while the "worst (individual)" is the worst training/testing classification accuracy among the 18 materials. The testing classification accuracy for noisy data is summarised in Table 3.4. Referring to this table, the column "worst"/"average"/"best" reports the worst/average/best of the average testing classification accuracy of the 18 materials among the 10 tests. The column "Std" is the standard deviation of the average testing classification accuracy of the 18 materials in the 10 tests. The column "Worst individual (average)" reports the worst average testing classification accuracy for an individual among the 18 materials.

		Classification Accuracy (%)			
		Training		Testing	
#feature points	Classifier	Worst Individual	Avg	Worst Individual	Avg
3	1	100	100	80	96.9444
3	2	92.5	98.8889	80	96.3889
3	3	97.5	99.7222	90	98.6111
3	4	100	100	100	100
3	5	95	99.3056	90	98.0556
3	6	97.5	99.8611	90	99.1667
3	7	100	100	80	95.8333
3	8	100	100	90	99.4444
4	1	100	100	85	96.3889
4	2	97.5	99.8611	85	96.3889
4	3	97.5	99.5833	90	98.8889
4	4	100	100	100	100
4	5	87.5	99.0278	80	98.3333
4	6	100	100	90	98.6111
4	7	100	100	70	93.6111
4	8	100	100	100	100
5	1	100	100	85	96.1111
5	2	100	100	95	98.3333
5	3	97.5	99.8611	95	99.7222
5	4	100	100	100	100
5	5	97.5	99.8611	95	99.1667
5	6	100	100	100	100
5	7	100	100	40	89.7222
5	8	100	100	100	100

Table 3.3: Summary of classification performance of the 6 NN-based classifiers, kNN classifier and Naive Bayes Classifier under noise-free dataset. Classifier 1: one-against-all classifier, classifier 2: weighted one-againsy-all classifier, classifier 3: binary-coded classifier, classifier 4: parallel-structured classifier, classifier 5: parallel-structured classifier, classifier 6: tree-structured classifier. classifier 7: K-nearest neighbor classifier, 8: Naive Bayes classifier

#feature points	Classifier	Classification Accuracy (%) Worst	Avg	Best	Std	Worst Individual (Average)
3	1	88.3333	92.9722	96.3889	2.7212	59.5000
3	2	80.0000	85.2500	89.4444	3.4430	0
3	3	94.4444	96.6944	98.8889	1.4593	74.5000
3	4	90.0000	93.9722	96.3889	2.2056	51.5000
3	5	86.9444	90.1944	92.5000	1.9889	5.5000
3	6	96.3889	97.8611	99.1667	0.9679	80.5000
3	7	88.6111	93.8889	98.3333	3.3120	84
3	8	92.5000	93.5278	94.1667	0.5826	0.5000
4	1	78.6111	83.0000	86.6667	2.8315	0.5000
4	2	81.6667	85.3889	88.6111	2.3061	0
4	3	90.0000	93.9444	96.3889	2.1650	64.5000
4	4	91.1111	92.7778	93.8889	1.0273	2.5000
4	5	90.2778	93.1667	95.2778	1.9215	10.5000
4	6	95.5556	97.3611	99.1667	1.0499	75.5000
4	7	81.6667	86.6667	91.1111	3.3264	27
4	8	95.5556	97.5000	98.6111	1.0499	72
5	1	87.7778	93.0278	97.2222	3.2587	56.5000
5	2	92.2222	94.8333	97.5000	1.9245	43
5	3	96.6667	98.8611	99.7222	1.1066	94
5	4	94.7222	95.8056	96.6667	0.6879	25
5	5	92.5000	96.5833	98.8889	2.1120	73.5000
5	6	99.1667	99.7778	100.0000	0.2869	96.0000
5	7	83.3333	88.6389	92.7778	3.0588	39.5000
5	8	93.3333	93.9167	94.4444	0.4086	0.5000

Table 3.4: Noise: Summary of classification performance of the 6 NN-based classifiers, KNN classifier and Naive Bayes Classifier. Classifier 1: one-against-all classifier, classifier 2: weighted one-againsy-all classifier, classifier 3: binary-coded classifier, classifier 4: parallel-structured classifier, classifier 5: parallel-structured classifier, classifier 6: tree-structured classifier. classifier 7: K-nearest neighbor classifier, 8: Naive Bayes classifier

3.4.1.1 One-Against-All Classifiers

An NN with 3 layers as shown in Fig. 2.1 is employed to implement the one-against-all classifier. The number of hidden nodes was chosen to be 30 and the transfer function of hidden nodes was chosen to be a logarithmic sigmoid transfer function.

Referring to Table 3.3, it can be seen that the average testing classification accuracy (the 6^{th} column in Table 3.3) is about 96% for the one-against-all classifier using feature vector of 3 to 5 feature points. However, looking into the worst individual testing classification accuracy of individual material (the 5^{th} column in Table 3.3), the one-against-all classifier using 3 feature points offers 80% classification accuracy while the one-against-all classifier using 4 or 5 feature points offers 85% testing classification accuracy in the worst case. It suggests that the feature vector of 3 feature points may not work well with the one-against-all classifier.

Considering the case that the test data subject to Gaussian white noise, it can be seen from Table 3.4 that the average classification accuracy (the 4^{th} column in Table 3.4) of the one-against-all classifiers subject to noisy data has declined to about 92%, 83% and 93% for 3, 4 and 5 feature vectors, respectively. The classifier with 4 feature points performs the worst when noise exists. Also, it is found that some individual materials are very sensitive to noise leading to a very low worst individual classification accuracy (the 7^{th} column in Table 3.4).

3.4.1.2 Weighted One-Against-All Classifiers

An NN with 3 layers is employed to implement the weighted one-against-all classifier. The elements of the weighting vector $\mathbf{w}_i$ were all chosen to be $\lceil i - 9.5 \rceil$, $i = 1, 2, \ldots, 18$. The number of hidden nodes was chosen to be 30 and the transfer function of hidden nodes was chosen to be hyperbolic tangent sigmoid transfer function.

Referring to Table 3.3, the average testing classification accuracy is about 96% for the weighted one-against-all classifier using the feature vector of 3 or 4 feature points. However, the average testing classification accuracy is improved to about 98% for the feature vector of 5 feature points. Looking into the worst individual testing classification accuracy, the weighted one-against-all classifier using feature vector of 3 feature points offers 80% while the weighted one-against-all classifier using 4 or 5 feature points offers 85% and 95% testing classification accuracy, respectively, in the worst case. Similar conclusion that the feature vector of 3 feature points may not work effectively can be drawn.

Referring to Table 3.4, the average performance of weighted one-against-all classifier under noisy data has declined to about 85%, 85% and 95% respectively. Similar observation is found as in the results from one-against-all classifiers as the same mechanism is used on both one-against-all and weighted one-against-all classifiers.

3.4.1.3 Binary-Coded Classifiers

An NN with 3 layers as shown in Fig. 2.1 is employed to implement the binary-coded classifier. The number of hidden nodes was chosen to be 30 and the transfer function of hidden nodes was chosen to be logarithmic sigmoid transfer function.

Referring to Table 3.3, the average testing classification accuracy for the binary-coded classifier with feature vector of 3 or 4 feature points is 98% while with feature vector of 5 feature points is about 100%. The worst individual testing classification accuracy is 90% for binary-coded classifier with feature vector of 3 and 4 feature points but is improved to 95% with feature vector of 5 feature points. Comparing with the one-against-all and the weighted one-against-all classifier, the classification performance of binary-coded classifier is less sensitive to the number of feature points.

Referring to Table 3.4, the average classification performance of binary-coded classifier under noisy data can be observed. The binary-coded classifier is able to offer a relatively higher performance compared with the one-against-all and weighted one-against-all classifiers. Corresponding to the number of feature points as 3, 4 and 5, the average classification accuracy can achieve about 97%, 94% and 99%, respectively. It is again showing that the dataset with 4 feature points produces the worst result. It is observed that worst average individual classification accuracy is much improved compared with the previous two classifiers, especially for the case with 5 feature points which can achieve 94%.

3.4.1.4 Parallel-Structured Classifiers

In the parallel-structured classifier, all NNs are with 3 layers where the number of hidden nodes was chosen to be 10 and the transfer function of hidden nodes was chosen to be logarithmic sigmoid transfer function. Compared with the NNs used in the above classifiers, the number of hidden nodes is significantly reduced, which supports the comment in Table 3.1 that the complexity of NN is relatively lower.

Referring to Table 3.3, the individual training and testing classification accuracy are all 100% irregardless of the number of feature points used. Of all classifiers, the parallel-structured classifier offers the best classification performance. Based on the average classification accuracy, it suggests that 3 feature points are sufficient for classification purposes.

Referring to Table 3.4, the performance of parallel-structured classifier under noisy data can be observed. It can be seen that the parallel-structured classifier is still able to offer a tolerable performance. Corresponding to the number of feature points as 3, 4 and 5, the average classification accuracy can achieve about 94%, 93% and 96%, respectively. The average classification performance is not as good as but comparable to that of the binary-coded classifier. Also, it can be seen from the worst average individual classification accuracy that some individual materials are very sensitive to noise.

3.4.1.5 Weighted Parallel-Structured Classifiers

In the parallel-structured classifier, all NNs are with 3 layers where the number of hidden nodes was chosen to be 15, the transfer function of hidden nodes was chosen to be hyperbolic tangent sigmoid transfer function and $G = 3$. The weighting vectors $\mathbf{w}_i$ were chosen as follows.

Feature vector of 3 feature points:

$$\mathbf{w}_i = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}, i = 1, 4, 7, 10, 13, 16.$$

$$\mathbf{w}_i = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}, i = 2, 5, 8, 11, 14, 17.$$

$$\mathbf{w}_i = [1 \quad 1 \quad 1]\,, i = 3, 6, 9, 12, 15, 18.$$

Feature vector of 4 feature points:

$$\mathbf{w}_i = [-1 \quad -1 \quad -1 \quad -1]\,, i = 1, 4, 7, 10, 13, 16.$$

$$\mathbf{w}_i = [1 \quad 1 \quad -1 \quad -1]\,, i = 2, 5, 8, 11, 14, 17.$$

$$\mathbf{w}_i = [1 \quad 1 \quad 1 \quad 1]\,, i = 3, 6, 9, 12, 15, 18.$$

Feature vector of 5 feature points:

$$\mathbf{w}_i = [-1 \quad -1 \quad -1 \quad -1 \quad -1]\,, i = 1, 4, 7, 10, 13, 16.$$

$$\mathbf{w}_i = [1 \quad 1 \quad 1 \quad -1 \quad -1]\,, i = 2, 5, 8, 11, 14, 17.$$

$$\mathbf{w}_i = [1 \quad 1 \quad 1 \quad 1 \quad 1]\,, i = 3, 6, 9, 12, 15, 18.$$

Referring to Table 3.3, it can be seen that the weighted parallel-structured classifier with feature vector of 5 feature points offers the best average testing classification accuracy of about 99% with the worst individual classification accuracy of 95%. Although the weighted parallel-structured classifier with feature vector of 3 or 4 feature points does not have a bad performance with an average testing classification accuracy of about 98%, the worst individual testing classification accuracy is 90% for 3 feature points and 80% for 4 feature points.

Referring to Table 3.4, the performance of weighted parallel-structured classifier under noisy data can be observed. The weighted parallel-structured classifier is able to offer tolerable average classification accuracy of about 90%, 93% and

97%, corresponding to 3, 4 and 5 points of feature vectors, respectively.

3.4.1.6 Tree-Structured Classifiers

In the tree-structured classifier, all NNs are with 3 layers where the number of hidden nodes was chosen to be 20 for the group classifier, 5 for each sub-classifier, the transfer function of hidden nodes was chosen to be logarithmic sigmoid transfer function for both the group classifier and sub-classifiers. The number of sub-classes is chosen to be $G = 3$. Compared with the NNs used in the above classifiers, the number of hidden nodes in the sub-classifier is much smaller as only 3 sub-classes need to be handled.

Referring to Table 3.3, the tree-structured classifier with feature vector of 5 feature points offers 100% training and testing classification accuracy while the one with 3 or 4 feature points offers about 99% testing classification accuracy and the worst individual testing classification accuracy of 90%.

It can be seen from Table 3.4 that the tree-structured classifier under noisy data is still able to offer a relatively high classification accuracy. Corresponding to 3, 4 and 5 points of feature vectors, the average classification accuracy of about 98%, 97% and 100%, respectively, can be achieved. Among all NN-based classifiers, the tree-structured classifiers are more robust to the noisy input.

3.4.2 Traditional Classifiers

In order to show the superiority and adaptability of the NN-based classifiers, two traditional classifiers, namely kNN classifier and the naive Bayes classifier, are employed to accomplish the classification of the 18 materials using the features vectors of 3, 4, and 5 points.

3.4.2.1 K-Nearest Neighbor Classifier

In this experiment, the classification accuracy for the kNN classifiers with feature vectors of 3 to 5 feature points was achieved by fixing the value of k for kNN classifier to 1.

From Table 3.3, it can be seen that the average classification accuracy for the training dataset is 100% for 3, 4 and 5 points of feature vectors. However, when the test dataset is considered, the kNN classifiers with 3 feature points can achieve average classification accuracy of about 96%, which is higher than that of the kNN classifiers with 4 and 5 feature points, which can achieve only 94% and 90% of average classification accuracy.

From Table 3.4, it can be found that the average testing classification performance of the kNN classifiers with noisy dataset has declined to some extent. The best average classification accuracy of about 94% is obtained for the feature vector of 3 points while the average classification accuracy is dropped to about 87% and 89% for the kNN classifiers with the feature vector of 4 and 5 points, respectively. It is interestingly observed that the worst individual average classification accuracy is able to achieved 84% for feature vector of 3 points while about 27% and 40% are achieved for feature vector of 4 and 5 points, respectively.

3.4.2.2 Naive Bayes Classifier

The classification accuracy for the naive Bayes classifier with feature vector of 3 to 5 points is summarised in Table 3.3. It can be seen from the table that the average classification accuracies of the naive Bayes classifier for the training dataset can achieve 100% for 3, 4 and 5 points of feature vectors while the average classification accuracies for the test dataset are 99%, 100% and 100%.

When noise is considered in the test dataset, the classification performance is given in Table 3.4. The average classification accuracies achieved for 3 to 5 feature points are about 94%, 98% and 94%, respectively. However, the worst individual classification accuracy reveals that the naive Bayes classifier is sensitive to noise.

3.5 Discussion

Giving an overall picture of the classification performance, Table 3.3 summarises the overall classification performance of the 6 NN-based classifiers and two traditional classifiers under the original dataset and Table 3.4 summarises the overall classification performance under noisy test dataset.

It can be seen from Table 3.3 that in general the classifiers with 5 feature points perform better in terms of the worst individual training and testing classification accuracy, and the average training and testing classification accuracy. When 5 feature points are considered, the parallel-structured, tree-structured classifier and naive Bayes classifer are able to offer the training and testing classification accuracy of 100%. The second best is the binary-coded classifier which is able to offer the training and testing classification accuracy around 99%. The worst one is the one-against-all classifier which is only able to offer a testing classification accuracy around 96%. When considering the kNN classifier, the overall classification accuracy for the training dataset is 100%. However, among all classifiers, the kNN classifier offers the worst classification accuracy for the test dataset.

Under the noisy test dataset, referring to Table 3.4, in general, the classification performance declines for all classifiers. The overall average classification accuracy drops below 90% for weighted one-against-all classifier when feature vector of 3 points is employed; for one-against-all classifier, weighted one-against-all classifier and kNN classifier when feature vector of 4 points is employed; for kNN classifier when feature vector of 5 points is employed. It is observed that majority of classifiers can obtain better classification accuracy when feature vector of 5 points is employed. By looking into the details, it can be seen that the tree-structured classifier can obtain the best average classification accuracy. In particular, when feature vector of 5 points is employed, the tree-structured classifier is able to achieve overall average classification accuracy of 99.7778%, outperforming the other classifiers. It can also be seen that the tree-structured classifier demonstrates consistent classification performance subject to noisy input with the smallest standard deviation among all classifiers. The second best is the binary-coded classifier which can obtain the overall average classification accuracy of 98.8611% but the standard derivation is more or less 5 times higher than that of the tree-structured classifier. The worst one is the kNN classifier which can obtain the overall average classification accuracy of 88.6389% with a significantly higher standard deviation. It is interestingly found that the parallel-structured classifier is less sensitive to the number of feature points used, which is able to offer more or less the same overall average classification accuracy regardless of the number of feature points subject to the original and noisy datasets.

From the summary tables, it can be concluded that the binary-coded classifier and tree-structured classifier are more suitable for the application of material classification when feature vector of 5 points are used.

3.6 Conclusion

The research conducted has introduced 6 neural-network-based classifiers (namely one-against-all, weighted one-against-all, binary coded, parallel structured, weighted parallel structured, tree-structured classifier) and two traditional classifiers (namely k-nearest neighbor classifier and naive Bayes classifier) to deal with a material classification problem where data was collected from a robot finger installed with tactile sensors. In total 18 materials have been considered in the experiment. The properties of each classifier have been discussed and its mechanism of performing classification has been detailed. To perform the classification, feature vectors of size 3, 4 and 5 are extracted for each material. Supervised learning approach has been adopted to train the neural-network-based classifier, kNN classifier and naive Bayes classifier for the classification of 18 materials. The performance of each classifier has been fully investigated and compared with each other in terms of classification accuracy. In the case using original dataset, the results show that the parallel-structured classifier produces the best performance among all 8 classifiers when 3, 4 and 5 feature points are used. However, under the case of dataset subject to noise, the tree-structured classifier has achieved the best performance among all the classifiers when 3, 4 and 5 feature points are used. In the future, some more advanced techniques such as variable-parameter neural network and fuzzy support vector machines can be applied to further improve the classification accuracy and robustness property.

Chapter 4

Variable Weight Neural Networks and Classification Problems

4.1 Introduction

In this chapter, we consider two real-life applications which are material surface classification and epilepsy seizure phases classification. In the application of surface material classification, we develop classifiers to recognise the surface material of an unknown object from 18 classes. In the application of epilepsy seizure phases classification, classification of epilepsy signals is considered using real clinical data. We will develop classifiers which are able to classify the 3 seizure phases namely seizure-free, pre-seizure and seizure phases. Both applications demonstrate a huge potential to be applied in domestic and industrial tasks. In both of the applications, we will employ the VWNN to implement the classifiers. The classification performance will be compared with some traditional classifiers such as feedforward-neural-network, naive Bayes and kNN classifiers.

Their robustness will be tested using noise-contaminated data.

The objective of the research conducted in this chapter is to improve the generalization capabilities and also robustness of the neural network by proposing the novel neural network-based classifier. One of the main limitations of the traditional neural network when dealing with large datasets is the fact that a large number of hidden neurons would be required to tackle this problem, in this chapter we use a different network to predict the weights for the classifier and as a result we can have an infinite number of networks, this is the main difference between VWNN and other NNs. Particular interest is being paid to its application to the material surface and epilepsy seizure phase classification. The benefits of improving the generalization capabilities of this classifier is very significant due to the implications that it would have in various industries such as the medical and industrial fields. This is the main reason behind the research being carried out in the chapter, the VWNN technique is relatively well known but the main contribution of this chapter is in its application to surface classification and epilepsy seizure phase classification.

The chapter is organised as follows. Section 4.2 introduces the VWNN and explains how it works. The method used to implement the VWNN classifier to the material and epilepsy phase classification problems is given in Section 4.3 with a presentation and discussion of results obtained in Section 4.4. Section 4.5 gives the conclusion.

4.2 Variable-Weight Neural Networks

A feed-forward fully-connected neural network is a network with static weights which processes all input using the same connection weights between layers. A detailed description of the feed-froward NN is written in 2.2.1 of the thesis. Although it has been shown that it is a universal approximator, it requires a sufficiently large number of hidden nodes to offer an acceptable performance. Considering the case when the number of input data is large, the number of hidden nodes will have to be large in order to maintain the learning and generalization capability. However, a large number of hidden layers is not favourable to both hardware and software implementations due to the increase of computational demand. Using a small number of hidden nodes will definitely offer advantages rather than disadvantages in terms of implementation costs. However, it will degrade the learning and generalization capability of the neural network resulting in a poor performance.

A VWNN is a neural network with dynamic weights which is good in handling a large dataset. Assuming that the large dataset is divided into a number of small sub-datasets, a small neural network (with small number of hidden nodes) will work well. The VWNN works based on the concept that different connection weights are employed by the neural network according to the network input. Consequently, the VWNN seems to consists of infinite number of neural networks and each individual input is processed by an individual neural network.

A three-layer VWNN is shown in Fig. 4.1. The tuning neural networks (NN_1 and NN_2) will provide connection weights $w_{ji}^{(1)}$ and $w_{ji}^{(2)}$ and bias weights $b_j^{(1)}$ and $b_j^{(2)}$ to a three-layer feed-forward fully connected neural network according to the input $\mathbf{x}'(t)$ which consists of some selected features from $\mathbf{x}(t)$. The neural network will process the input $\mathbf{x}'(t)$ according to the provided connection weights. This concept can be generalised to a VWNN with any number of hidden layers.

A block diagram of a general VWNN is shown in Fig. 4.2 which consists of 2 traditional neural networks, namely tuning and tuned neural networks. The input $\mathbf{x}(t)$ will be selected by a pre-determined constant selection matrix $\mathbf{S} \in R^{n'_{n_{in}} \times n_{n_{in}}}$ such that $\mathbf{x}'_{n_{in}}(t) = \mathbf{S}\mathbf{x}_{n_{in}}(t)$ where $n'_{n_{in}} \leq n_{n_{in}}$. For example, considering $\mathbf{x}'_{n_{in}}(t) = \begin{bmatrix} x'_1(t) \\ x'_2(t) \end{bmatrix}$, $\mathbf{x}_{n_{in}}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$ and $\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, we have $\mathbf{x}'_{n_{in}}(t) = \mathbf{S}\mathbf{x}_{n_{in}}(t) = \begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix}$ which selects $x_1(t)$ and $x_3(t)$ as the input of the turning neural network. The tuning neural network will produce output weight vector $\mathbf{W}(t)$ consisting of all connection weights of the tuned neural network. The tuned neural network will then use $\mathbf{W}(t)$ to process the input $\mathbf{x}(t)$. As a result, it seems like an individual input $\mathbf{x}(t)$ is processed by an individual neural network to produce an output $\mathbf{y}(t)$.

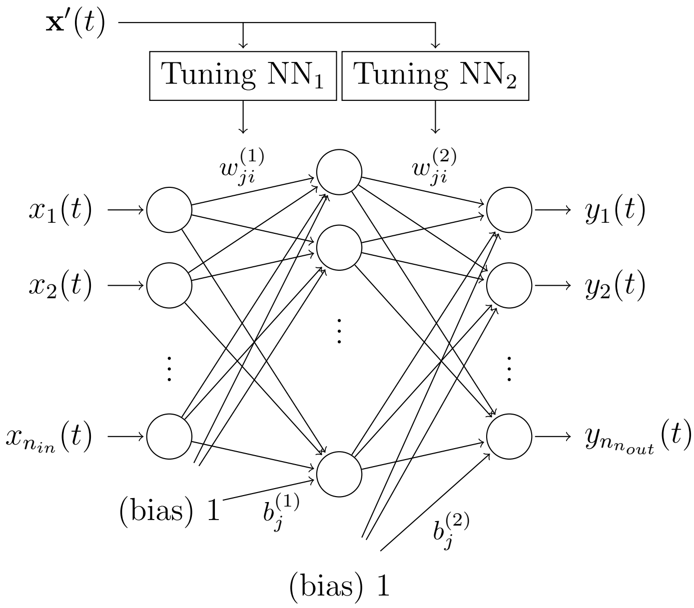

Figure 4.1: A three-layer variable-weight neural network.

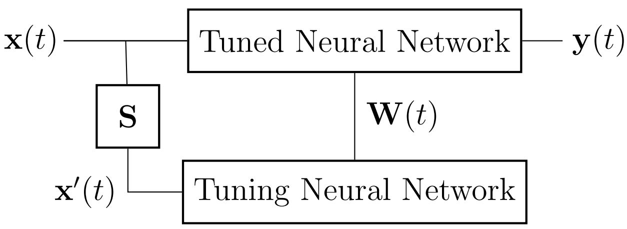

Figure 4.2: A block diagram of variable-weight neural network.

4.3 Method

In this section, we present a detailed description of the method used in applying the proposed VWNN classifier to handle two applications. The first application is the classification of materials using the data collected by a robotic finger which is described in Section 3.3 in Chapter 3 of the thesis. The second application which is described below in Section 4.3.2 is the classification of epileptic seizure phases using real clinical data.

4.3.1 Material Classification

The proposed VWNN is employed in the material classification problem to implement a classifier to recognise the 18 materials using the feature vectors of 3, 4, and 5 points. Fig. 4.3 shows the structure of classifier consisting of a VWNN with n_{in} inputs (the number of feature points) and one output.

In this experiment, the dataset is divided into training dataset consisting of 40 sets of data for each material and test dataset consisting of 20 sets of data for each material. Supervised learning was employed to train the VWNN classifier according to the class labels shown in Table 3.2 in chapter 3 of the thesis.

We have tried different combinations of transfer functions, number of hidden nodes and hidden layers in this study. In the following, only this combination can achieve the best classification accuracy. The overall network is 6 hidden layers structure, the number of hidden nodes are 3, 4, 6, 4, 3, 4, respectively. The first two layers's transfer function use 'tansig' function and other layers are 'linear' function, and the 3^{rd} and 4^{th} layers are VWNN layer, which tuned by two ordinary networks using 'tansig' transfer function. The linear transfer function is used in the output layer of all classifiers, and number of output nodes is 1. The VWNN classifier was implemented on Matlab and Levenberg-Marquardt back-propagation was used to train the classifiers by minimizing the mean square error.

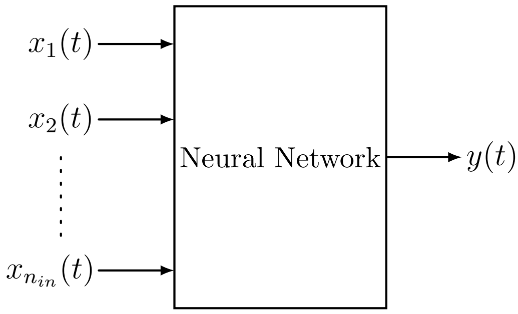

Figure 4.3: NN-based classifier for materials.

For comparison purposes, traditional NN, kNN and naive Bayes classifiers were employed as classifiers for this application. To test the robustness of the classifiers, the test dataset extracted from raw data contaminated by Gaussian white noise with variance of 0.005 and 0.01 was considered. Each classifier was tested 10 times using the noisy test dataset.

4.3.2 Epilepsy Seizure Phase Prediction

Epilepsy, which is characterised with its ability to instantiate recurrent seizures (an interruption of normal brain functions) which are unforeseen in nature is a very common and significant neurological disorder caused by a sudden discharge of cortical neurons [30, 82]. Epileptic seizures are classified as either partial (involving focal brain regions) or generalised (where it involves a widespread region of the brain across both hemispheres) [10]. The length of time for the seizure occurrence varies from a few seconds up to a minute with some of the effects including momentary lapse of consciousness for the sufferer of the seizure [10]. A complete loss of consciousness occurs when the epileptic activity involves both the cortical and subcortical structures of the brain and this occurrence is known as an absence seizure.

The unexpected nature of these seizures has proven to have an adverse effect on the quality of life for those who are suffering from them. The impact is most prevalent in the formative stages of a childs life as we see an increase in the requirements for special education and also a higher incidence of below-average school performance [62, 82]. It also proves life-threatening in situations where the sufferer is isolated at the time of its occurrence and there is no experienced or medical help on hand to alleviate the situation. Therefore having an accurate understanding or predictive model for the pre-seizure phase (the transition towards an absence seizure occurrence) is a very vital task as it would provide the sufferers and their carers enough notice of the upcoming seizure so they could prepare themselves and dampen the impact of the seizure occurrence.

Absence seizures can be best characterised by the spike-and-wave discharges (SWDs) which are as a result of synchronised oscillations in the thalamocortical networks of the brain [38, 89]. The classification process of EEG signals consists of two main parts which are feature extraction and classification. In the literature, there are a wide range of available feature extraction methods which range from the traditional methods to the non-linear methods. Traditional methods include the fourier transform and also spectral analysis whilst the non-linear methods include Lyapunov exponents [52, 82], correlation dimension [82] and similarity [97]. After feature extraction has been implemented to the raw data, the extracted features are then used and applied to the pre-determined classification technique.

There are a wide range of classification techniques for EEG classification in the literature, examples of these include the artificial neural network and also the neuro-fuzzy systems [40, 61, 131].

For this particular problem of accurately classifying and thereby predicting the onset of an epileptic seizure, the extracted features are applied to various classifiers (kNN, naive Bayes, SVM and IT2FSVM) with the main aim of being able to recognise and distinguish between the 3 seizure phases (seizure-free, pre-seizure and seizure phase). The raw data obtained for the simulations being carried out were obtained from the Peking University by the aid of 10 patients who were suffering from absence epilepsy, their ages ranging from 6 to 21 years old. The study has been approved by the ethics committee of Peking University Peoples Hospital and the patients all signed documents in consent of their clinical data being used for research purposes. The EEG data (which was sampled at a frequency of 256 Hz with the aid of a 16-bit analogue-to-digital converter and then filtered within a frequency band of 0.5 to 35 Hz) was recorded by the Neurofile NT digital video EEG system using a standard international 10-20 electrode placement (Fp1, Fp2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T3, T4, T5, T6, Fz, Cz, and Pz). There are 3 sets of EEG signals which are extracted from the 3 seizure phases (seizure-free, pre-seizure and seizure) to obtain 112 2-second 19-channel EEG epochs from 10 patients for each dataset. The timing of the onset and offset in the spike-wave discharges (SWDs) were identified by a neurologist and these SWDs were identified to be large amplitude 3-4Hz discharges with a spike-wave morphology typically lasting above a second in duration. The criteria for determining the different seizure phases (seizure-free, pre-seizure and seizure) are that there is an interval between the seizure-free phase and beginning of the seizure phase which is greater than 15 seconds, the interval is between 0 to 2 seconds before the occurrence of the seizure and that the interval occurs during the first 2 seconds of the absence seizure.

4.3.2.1 Feature extraction

The feature extraction procedure is very vital in the classification process as it obtains the relevant characteristics and information from a large dataset (EEG signals in this instance), this has the knock-on effect of simplifying the dataset and also reducing the effect of redundant datapoints that have little or no effect in the classification of the dataset. This is a very important step in improving the performance of the classifier as classification is easier when the classifier is subject to fewer datapoints. For the EEG case being undertaken, there are 19 columns of signal output. The 19 columns represent signals that were drawn from 19 EEG sensors with each column containing 100 sample points. The purpose of the feature extraction being carried out here is to extract the relevant feature points from the 19 x 100 dataset and thereby reducing the dimensionality.

Research into the existing literature provides evidence to suggest that the 19 channels of the EEG data vary in importance with regards to classification, it was observed that some of the channels have a lesser impact on the classification of the EEG and the exclusion of these channels has been investigated in [3, 110]. Both studies have discovered that some of the electrodes (F3, Fz, F4, C3 and Cz) are the most significant ones for the classification between the seizure-free and seizure patients and the remaining electrodes are found to have relevant information for the classification between the different seizure phases.

We utilise the relevant channels for classification (i.e channel 1, 2, 3, 4, 5, 6, 11, 12, 13, 14) for the simulations carried out in this paper. For each of the channels, a feature vector containing the time-domain and frequency-domain components of the dataset is created. The first part of the feature vector comprises of computations in the time-domain such as the standard deviation, second order norm, third order norm, fourth order norm, absolute sum, maximum value and minimum value of the 100 sample points from each channel. The second part is comprised of computations in the frequency domain such as the mean frequency, maximum frequency, minimum frequency, standard deviation of frequency, windowing filtered mean frequency and windowing filtered maximum frequency of each chosen channel will form the second part of the feature vector.

A problem that arises from these computations would result in a large vector which would be difficult to classify, this is solved by implementing principal component analysis (PCA) to reduce the number of dimensions in the feature vector. After this dimensionality reduction method has been implemented, we finally have 45 points which form the feature vector, this feature vector is then applied to the pre-determined classifiers.

4.3.2.2 Implementation

A tree-structured classifier implemented by the proposed VWNN is employed to classify 3 classes of Epilepsy signals (seizure-free, pre-seizure and seizure phases) using the feature vector achieved in the previous section. Amongst the three epilepsy seizure phases, the pre-seizure and seizure-free phases are very similar in terms of the EEG pattern and are therefore difficult to separate. The seizure phase is distinctly different from these two phases and is therefore relatively easy to classify an input pattern as belonging to the seizure phase when compared to the pre-sezire and seizure-free phases. For this reason, a tree-structured VWNN classifier consisting of 2 45-input-single-output VWNNs and a class determiner (as is shown in Fig. 4.4) is proposed to deal with this classification problem. The 1^{st} VWNN is used to determine if the testing sample belongs to class 3 (seizure phase), if not, we use the 2^{nd} VWNN to determine if the testing sample belongs to class 1 (seizure-free phase) or class 2 (pre-seizure phase). The classifier will determine the final class according to the rules as shown in Table.4.4.

In this application, we have tried different combinations of transfer functions, number of hidden nodes and hidden layers in this study. The following combination can achieve the best classification accuracy. For the 1^{st} VWNN network as shown in Fig. 4.4, the tuned NN has 45 inputs, 4 hidden layers with 25, 4, 8 and 5 hidden nodes and one output node. The transfer functions corresponding to the 4 hidden layers are hyperbolic tangent sigmoid, hyperbolic tangent sigmoid, logarithm sigmoid function and logarithm sigmoid function, respectively. Linear function and logarithm sigmoid function are used in the input and output layers, respectively. The tuning NN has 45 inputs, 2 hidden layers with 25 and 4 hidden nodes and 32 output nodes. The first 3 layers, i.e., the input and the 2 hidden layers, are common to the tuned NN. The output layer uses hyperbolic tangent sigmoid function as the transfer function. The outputs of the tuning NN provide the variable weights to the connections between the 3^{rd} and 4^{th} hidden layers of the tuned NN.

The tuned NN of the 2^{nd} VWNN as shown in Fig. 4.4 has 45 inputs, 4 hidden layers of 35, 5, 8 and 5 hidden nodes and one output node. The transfer functions corresponding to the 4 hidden layers are hyperbolic tangent sigmoid, hyperbolic tangent sigmoid, logarithm sigmoid function and logarithm sigmoid function, respectively. Linear function and logarithm sigmoid function are used in the input and output layers, respectively. The tuning NN has 45 inputs, 2 hidden layers with 35 and 5 hidden nodes and 40 output nodes. The first 3 layers, i.e., the input and the 2 hidden layers, are common to the tuned NN. The output layer uses hyperbolic tangent sigmoid function as transfer function. The outputs of the tuning NN provide the variable weights to the connections between the 3$^{\text{rd}}$ and 4$^{\text{th}}$ hidden layers of the tuned NN.

For comparison purposes, traditional NN, kNN and naive Bayes classifiers were employed as classifiers for this application. To test the robustness of the classifiers, the test dataset extracted from the raw data contaminated by Gaussian white noise with variance of 0.05 to 1 was considered. The NN classifier has the same structure as the VWNN classifier as shown in Fig. 4.4 but the VWNNs are replaced by the traditional NNs. The transfer function used in the 3$^{\text{rd}}$ layer of the traditional NN which has the same number of hidden nodes as that of the VWNN. Each classifier was tested 10 times using the test dataset subject to noise of different levels.

4.4 Results and Discussion

In this section we present and give a detailed discussion of the results obtained from the implementation of the VWNN to the material classification and epilepsy phase classification problems. A comparison is also made to other traditional classification techniques to gain an insight into the effectiveness of the proposed VWNN classifier.

4.4.1 Material Classification

The training and testing classification results of the VWNN, traditional NN, kNN and naive Bayes classifiers are summarised in Table 4.1. In this table, the worst and average classification accuracies for both training and test datasets are shown. The worst classification accuracy is the worst individual accuracy in the 18 materials while the average classification accuracy is the average individual accuracy of 18 materials.

Referring to this table, it can be seen that all classifiers perform well, achieving 100% of classification accuracy for training dataset. For test dataset, the naive Bayes classifiers with 3, 4 and 5 feature points offer the best average classification performance of 99.4444%, 100% and 100%, respectively. The proposed VWNN classifiers with 3, 4 and 5 come second offering the average classification performance of 98.6111%, 98.6661% and 99.1667%. All the remaining classifiers offer an average classification performance of less than 97%. Among all remaining classifiers, the kNN classifier with 5 feature points offers the worst average performance of 89.7222% and its worst individual classification accuracy is 70%.

The performance of the VWNN for this particular application shows its viability and potential as a classifier, apart from the Nave Bayes classifier, we observe that the VWNN outperforms all other classifiers significantly as its classification accuracy is at least 2% greater than all the other classification techniques that are used for comparison. The distinct advantage that the VWNN has over all other techniques is its robustness and flexibility as the VWNN can be applied to any range of input data without having an adverse impact in the performance, this is due to the fact that a separate NN selects the weights used for the tuned

NN. This makes the VWNN a very competent classifier for the material surface classification application as it would not just be able to recognise and classify the present materials but will have the ability to recognise additional material types that could be added to the system without having a significant reduction in its classification accuracy.

The testing classification results for the test dataset subject to Gaussian white noise with variance of 0.005 and 0.01 are summarised in Table 4.2 and Table 4.3, respectively, which provide the statistical information including the average classification accuracy (the average of the average classification accuracy of the 18 materials of the 10 times of tests), worst classification accuracy (the worst average classification accuracy of the 18 materials among the 10 times of tests) and best classification accuracy (the best average classification accuracy of the 18 materials among the 10 times of tests), standard deviation of the 10 times of tests and the average of the worst individual classification accuracy among 18 materials.

In machine learning, the error due to the bias is the difference between the predicted value of the model and the correct value that we are trying to predict. The error due to the variance is the variability of a model prediction for a given datapoint [35]. The objective of machine learning is to minimise both the bias and the variance of the proposed model. This is difficult in practise as an increase in complexity of the classify may enable it to better fit the data but this may likely result in overfitting where the classifier also models the noise inherent in the input data and therefore generalises poorly with unseen data. A balance between the bias and variance is therefore required in order to optimise the classification accuracy of the proposed classifier. The VWNN is able to have a superior classification accuracy to the traditional classifiers due to the fact that a better bias-variance tradeoff is attained by the VWNN structure when compared to that of the traditional classifiers. This is due to the fact that the VWNN has the ability to vary the structure of the classifier with each new input and is therefore better suited to model the complex input patterns inherent in the feature extracted input from the 18 different materials. As a result of this, the VWNN is able to attain a high level of classification accuracy (low bias) with a relatively low level of complexity (low variance) when compared to the traditional classifiers. The VWNN has low structural complexity when compared to the traditional NN because the structure has a separate NN producing the weights for the main NN. This enables it to have better ability to interpret the data with a less complex NN structure when compared to the traditional NN.

It can be seen from the tables that the classification performance of all classifiers degrade when the noise level increases. Considering the noise level of 0.005 in Table. 4.2, the VWNN classifier with 5 feature points, and naive Bayes classifier with 4 and 5 feature points offer the best average classification accuracy over 98%. We also see in the case of 5 feature points that the VWNN classifier has a worst individual accuracy of 68% which is better than all the traditional classifiers being used for comparison. The VWNN classifier also a has a standard deviation of 1.582 and therefore variance of 2.501 and this is lower than that of the traditional NN and kNN classifiers. This shows that the VWNN classifier has a low bias and low variance when subjected to an input with a noise level of 0.005. When the noise level increases to 0.01 as is shown in Table. 4.3, naive Bayes classifiers degrade their performance significantly compared with the VWNN classifier with 5 feature points. The VWNN classifier with 5 feature points is able to maintain its classification performance offering the best of the best average classification accuracy of 94.7222% while the kNN classifier with 3 feature points comes second offering 93.6111%.The worst individual accuracy for the VWNN can be seen to be higher than that of the traditional NN and NB, it also has a standard deviation of 3.650 and variance 13.323 that is lower than that of the kNN and traditional NN classifers. The results show that the VWNN classifier performs better and has a better generalisation performance which means that the classifier is able to attain a better bias-variance tradeoff than the traditional classifiers.

Based on the above discussion, the VWNN classifier with 5 feature points offers the best classification performance with noise-free raw data. Under the noisy raw data, it is able to outperform the other classifiers in terms of worst, average and best classification accuracy suggesting that it has a comparatively superior capability to tolerating noise in the input. In a real world application there would always be an element of noise in the input dataset and this is the main reason why noise was applied to the input data to investigate what the effect would be on classification accuracy. It is in this instance that we are able to see the effect of the robustness and performance of the VWNN neural network as we see that it outperforms all other classification methods when the system has been subjected to a noise level of 0.01. This shows the suitability and superiority of the VWNN as a classifier when compared to the Nave Bayes, kNN and Traditional NN as it has a higher performance and shows a lot of promise when applied to raw data as it has a higher level of tolerance to noise without adversely affecting its performance. For the material classification problem, the cost of misclassification is very high as it can result in costly mistakes such as the inclusion of foreign and potentially harmful particles in the production cycle or in the case of a remote sensing application in war zones, the cost of misclassifiction can result in massive human casualties if the classifier is unable to detect the presence of explosive devices. The superior classification accuracy of the VWNN and therefore the significantly reduced probability of misclassification makes it a very suitable classifier to deal with this problem.

		Classification Accuracy (%)			
		Training		Testing	
#feature points	Classifier	Worst	Average	Worst	Average
3	1	100	100	90	98.6111
3	2	100	100	80	96.9444
3	3	100	100	80	95.8333
3	4	100	100	90	99.4444
4	1	100	100	90	98.6661
4	2	100	100	85	96.3889
4	3	100	100	70	93.6111
4	4	100	100	100	100
5	1	100	100	95	99.1667
5	2	100	100	85	96.1111
5	3	100	100	70	89.7222
5	4	100	100	100	100

Table 4.1: Summary of classification performance under noise-free dataset. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

		Classification Accuracy (%)				
#feature points	Classifier	Worst	Average	Best	Std	Worst Individual (Average)
3	1	91.3889	94.2222	97.2222	2.0916	50.0000
3	2	86.3889	92.3611	96.3889	3.1882	55.0000
3	3	88.6111	93.8889	98.3333	3.3120	84.0000
3	4	92.5000	93.5278	94.1667	0.5826	0.5000
4	1	88.6111	91.2222	93.0556	1.4722	0.0000
4	2	77.5000	82.3333	86.9444	3.1543	1.5000
4	3	81.6667	86.6667	91.1111	3.3264	27.0000
4	4	95.5556	97.5000	98.6111	0.9631	72.0000
5	1	93.6111	96.3889	98.3333	1.5817	68.0000
5	2	86.3889	92.9444	97.2222	3.5728	60.0000
5	3	83.3333	88.6389	92.7778	3.0588	39.5000
5	4	93.3333	93.9167	94.4444	0.4086	0.5000

Table 4.2: Summary of classification performance for the dataset subject to noise level of 0.005. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

		Classification Accuracy (%)				
#feature points	Classifier	Worst	Average	Best	Std	Worst Individual (Average)
3	1	58.6111	63.9167	68.0556	3.1284	0.0000
3	2	61.1111	68.6944	74.1667	4.1574	0.0000
3	3	77.7778	86.2500	93.6111	5.2281	50.5000
3	4	74.7222	77.9722	81.1111	2.2443	0.0000
4	1	61.9444	64.7500	67.2222	1.8301	0.0000
4	2	58.6111	66.6389	72.2222	4.6941	0.0000
4	3	55.8333	62.2778	66.9444	3.7609	0.0000
4	4	67.7778	71.6111	74.7222	2.1802	0.0000
5	1	84.1667	89.5000	94.7222	3.6503	34.5000
5	2	63.8889	72.9444	80.0000	5.4622	0.0000
5	3	72.5000	82.6111	89.7222	5.8274	38.5000
5	4	79.7222	81.0185	82.5000	1.5608	0.0000

Table 4.3: Summary of classification performance for the dataset subject to noise level of 0.01. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

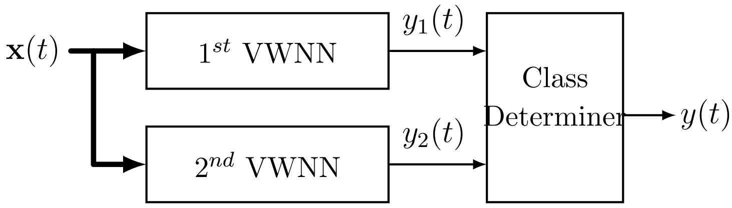

Figure 4.4: Tree-structure VWNN classifier for epilepsy.

$y_1(t)$	$y_2(t)$	$y(t)$
3	1	3
3	2	3
not 3	1	1
not 3	2	2

Table 4.4: Output classes of class determiner.

4.4.2 Epilepsy Seizure Phase Classification

The classification performance of all classifiers with the original data and data contaminated by noise level from 0.05 to 1 are shown below. The classification performance corresponding to different noise levels are summarised in Table 4.5 to Table 4.10.

	Classification Accuracy (%)			
	Training		Testing	
Classifier	Worst	Average	Worst	Average
1	100	100	80.0000	91.1111
2	100	100	73.3333	86.6667
3	100	100	23.3333	56.6667
4	41.4286	77.1429	33.3333	77.7778

Table 4.5: Summary of classification performance for EEG signals with original dataset. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)				
Classifier	Worst	Average	Best	Std	Worst Individual (Average)
1	85.5556	89.2222	94.4444	2.9801	75.0000
2	77.7778	85.0000	90.0000	4.0140	75.3333
3	51.1111	56.3333	61.1111	3.2735	20.3333
4	77.7778	78.3333	80.0000	0.7857	35.0000

Table 4.6: Summary of testing samples classification performance for EEG signal under dataset subject to noise level of 0.05. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)				
Classifier	Worst	Average	Best	Std	Worst Individual (Average)
1	80.0000	86.4444	92.2222	4.1869	67.0000
2	78.8889	85.7778	90.0000	3.6501	70.6667
3	52.2222	56.5556	61.1111	2.9232	20.3333
4	76.6667	78.8889	82.2222	1.8251	37.6667

Table 4.7: Summary of testing samples classification performance for EEG signal under dataset subject to noise level of 0.1. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)				
Classifier	Worst	Average	Best	Std	Worst Individual (Average)
1	78.8889	84.4444	90.0000	3.4978	58.6667
2	75.5556	84.1111	88.8889	3.8639	64.6667
3	53.3333	57.2222	62.2222	2.9614	20.6667
4	76.6667	79.0000	82.2222	1.8898	38.6667

Table 4.8: Summary of testing samples classification performance for EEG signal under dataset subject to noise level of 0.2. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)				
Classifier	Worst	Average	Best	Std	Worst Individual (Average)
1	81.1111	83.5556	85.5556	1.5948	54.6667
2	76.6667	81.2222	86.6667	3.6429	56.0000
3	52.2222	59.0000	64.4444	3.8665	24.6667
4	75.5556	78.2222	80.0000	1.5585	36.3333

Table 4.9: Summary of testing samples classification performance for EEG signal under dataset subject to noise level of 0.5. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)				
Classifier	Worst	Average	Best	Std	Worst Individual (Average)
1	77.7778	83.4444	85.5556	2.4525	54.6667
2	72.2222	78.3333	82.2222	3.9338	49.0000
3	50.0000	59.6667	66.6667	5.8187	23.3333
4	76.6667	77.7778	80.0000	1.3242	36.3333

Table 4.10: Summary of testing samples classification performance for EEG signal under dataset subject to noise level of 1. Classifier 1: VWNN classifier, classifier 2: traditional neural network classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

In Table 4.5, the worst and average classification accuracies for both training and test datasets are shown. The worst classification accuracy is the worst individual accuracy in the 3 classes while the average classification accuracy is the average classification accuracy of the individual classification accuracy of all 3 classes. It can be seen that the VWNN offers the best performance over the other 3 traditional classifiers methods evident by average training and testing classification accuracies of 100% and 91.1111%, respectively.

The fact that the VWNN outperforms the other existing classification methods (traditional NN, kNN and Nave Bayes) in all the simulations and when subjected to various levels of noise in the input data, it is fair to conclude that the VWNN is a very powerful tool with a strong suitability for this particular application and the flexibility of the VWNN to accept a wide range of input data will be very effective in dealing with the epilepsy seizure phase classification problem as this technique would enable to classifier to accept a wide range of patient data and have a good generalization ability that is not affected by this increase in the range of input data. The fact that the VWNN performs well under higher levels of noise further enhances its suitability as a classifier due to the fact that real-life data will not be smooth and will always be subject to noise in the system due to different factors such as levels of error in the measurements techniques.

Table 4.6 to Table 4.10 show the testing data classification performance of all the 4 classifiers with noisy data under noise levels of 0.05, 0.1, 0.2, 0.5, and 1. From these tables, it can be seen that the classification accuracy is in general decreasing when the noise level is increasing. Among the 4 classifiers, the VWNN classifier offers the best classification performance with the average classification accuracy in the range of 83.4444% of 89.2222% subject to different noise levels while kNN classifier performs the worst offering the average classification accuracies in the range of 50% to 53.3333%.

From the above discussion, it can be concluded that the VWNN classifier outperforms the traditional classifiers offering the best classification performance and robustness property.

4.5 Conclusion

In this chapter, we presented a novel neural network, the variable weights neural network, which demonstrates a great potential to cope with complicated classification problems. Different from the traditional NN, the weights of the VWNN change adaptively according to the characteristic of the input data thereby enhancing its learning and generalization capability. We have implemented classifiers using VWNNs for 2 real-life applications, i.e., material classification using robotic finger and epilepsy classification using clinical data, to verify the effectiveness of VWNN. From the results of these two applications, it has been shown that the VWNN classifier has demonstrated the best classification performance over the traditional neural networks, kNN method and Naive Bayes method when original input data are considered. Moreover, the VWNN classifier has demonstrated an outstanding robustness property towards noisy input data. In the future, we will keep improving the performance of VWNN and trying to find the best way to determine the structure of VWNN, for example, the number of hidden layers, the transfer function of each layer and the nodes of each layer.

Chapter 5

Classification of Epilepsy Seizure Phase using Interval Type-2 Fuzzy Support Vector Machines

5.1 Introduction

The application being considered in this chapter is the classification of the phases involved in the onset of an epileptic seizure, where the epilepsy signals obtained from the Electroencephalograph (EEG) using real clinical data is subjected to the novel classification technique [30, 82]. This is a very challenging classification problem as the EEG has multiple features and is also contaminated with noise and distortion [3, 110]. The classification technique is designed to differentiate between the 3 seizure phases namely seizure-free, pre-seizure and seizure. The early detection of seizure phases is a potentially life-saving application/research field and this is a major motivation for this research. The accurate classification/differentiation between the 3 seizure phases would give doctors and other healthcare professionals ample time to be able to prepare for the oncoming seizure. Therefore the main objective of the research carried out in this chapter is to propose an adequate classifier to deal with this problem. As a result of this, an interval type-2 fuzzy support vector machine (IT2FSVM) is being proposed to deal with this problem. The IT2FSVM will be utilised to differentiate between the 3 seizure phases. The IT2FSVM is proposed due to its superior ability at dealing with uncertainties and unbalanced data [79]. This therefore provides a higher level of classification accuracy than the traditional SVM and forms the basis for the implementation of this classifier. The classification performance of the IT2FSVM technique will be compared to some traditional classifiers including the kNN technique

To aid comprehension of the work done in this chapter, it is important to first read the background knowledge on the SVM which is in Section 2.2.2, the IT2FIS in Section 2.2.3, the type reduction method (KM algorithm) in Section 2.2.4 all in Chapter 2 of the thesis. The description of the epilepsy seizure phase classification problem is in Section 4.3.2 in Chapter 4 of the thesis with a description of the feature extraction method provided in Section 4.3.2.1. This chapter is organised as follows: In Section 5.2 the IT2FSVM structure is proposed with a detailed schematic to illustrate how it functions. Section 5.3 details how the IT2FSVM method is implemented to solve the classification problem. Section 5.4 presents and analyses the experimental results obtained from the application of the IT2FSVM method to the epilepsy seizure phase classification problem with a comparison to other existing methods followed by a discussion of the results obtained. Section 5.5 draws a conclusion.

5.2 Interval Type-2 Fuzzy Support Vector Machines (IT2FSVMs)

In this section, the mechanism of the IT2FSVM classifier is discussed. The standard SVM classifier is used for this hybrid classification mechanism which involves the merging of an IT2FIS with an SVM to form the IT2FSVM. The IT2FSVM can be characterised as a multiple-input-single-output classifier. The ability of the IT2FIS to handle uncertainty makes it very complementary to the SVM in solving difficult non-linear problems.

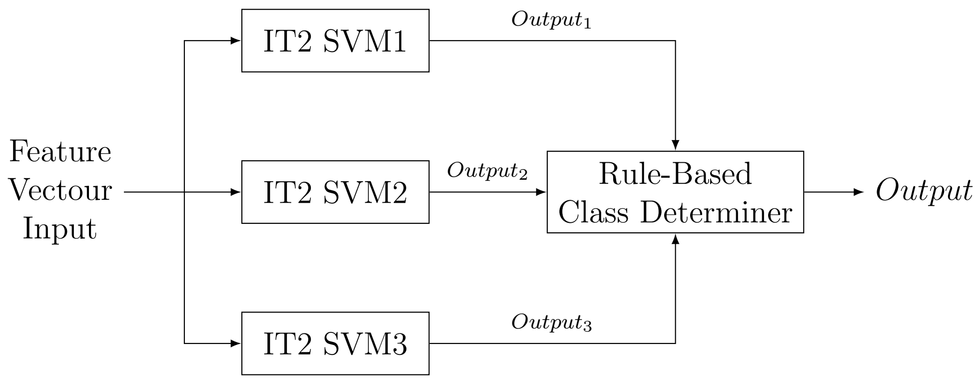

Figure 5.1: Block diagram of IT2FSVM.

The overall IT2FSVM architecture is shown in Fig. 5.1. The original EEG input is first subjected to feature extraction in order to extract relevant features from the dataset. The feature vector input is then fed into the IT2FSVMs which are represented by the IT2 SVM blocks in the diagram. As the hyperplane can only separate 2 classes, multiple SVMs are required as there are more than 2 classes in a classification problem. For the application in this chapter which is to differentiate between the epileptic seizure stages, multiple SVMs are required as there are three classes (seizure-free, pre-seizure and seizure). There are three

Case	$Output_1$	$Output_2$	$Output_3$	Final Class ($Output$)
1	-1	-1	-1 or 1	1
2	1	-1 or 1	-1	2
3	-1 or 1	1	1	3
4	1	-1	1	3
5	-1	1	-1	3

Table 5.1: Table showing the if-then rules used by the rule based class determiner system. Table showing the if-then rules. Class 1: Seizure-free, class 2: Pre-seizure, class 3: Seizure

IT2 SVM blocks in the diagram which are used to individually separate between the seizure phases. IT2 SVM 1 separates between the seizure-free and pre-seizure phases with the label "-1" indicating the input data belongs to the seizure-free class and label "1" indicating the input data belongs to the pre-seizure class. IT2 SVM 2 separates between the seizure-free and seizure phase with the label "-1" indicating the input data belongs to the seizure-free class and label "1" indicating the input data belongs to the seizure class. Finally, IT2 SVM 3 separates between the pre-seizure and seizure phase with the label "-1" indicating the input data belongs to the pre-seizure class and label "1" indicating the input data belongs to the seizure class. The output of the three IT2 SVM blocks are presented in $Output_1$ to $Output_3$. The output from the 3 IT2 SVM blocks are then fed into the rule-based class determiner in order to get the final classification output.

In the following paragraphs, a more detailed description of the IT2FSVM structure is provided. This will constitute of more technical details to aid in the potential reproduction of the simulations carried out for the research conducted in this chapter. The original EEG input data had a 19×100 vector input and feature extraction is used to reduce it to a 45-input feature vector which is used as the input of the IT2SVM. This reduces the dimensionality of the input data and thereby improves the classification performance. More details about the feature extraction method can be found in Section 4.3.2.1 in Chapter 4 of the thesis.

The IT2FSVM block consists of a feature vector input, 3 fuzzy rules each consisting of two SVMs associated with the lower and upper membership functions and a defuzzification block which is used to produce the final crisp output. In this block, the feature vector input is normalised and each input sample is assigned a membership grade based on the IT2 membership functions with the GA being used to optimise the shape of the IT2 membership functions. The objective function for the GA is to maximise the classification accuracy of the overall classifier architecture. This process can be further improved by tweaking the parameters of the genetic algorithm such as the number of iterations or the population size. A full list of the GA parameters can be found in Table. 5.2.

The SVMs are then trained with a section of the feature vector input after which unseen data can be applied in order to produce an output. Although the standard SVM is used for this purpose, the results can be optimised by adjusting the SVM parameters such as the regularization constant and kernel function. The output of the SVM for each of the fuzzy rules is then combined with the membership grade for that particular input to produce a fuzzy output. The number of fuzzy rules can be defined by any integer value but an increase in the number of fuzzy rules would lead to a slower convergence of training and also a higher computational cost of the system. In this chapter, there are three fuzzy rules employed to implement the IT2FSVM. Referring to Fig. 5.1, we have three

IT2 SVMs. Each IT2 SVM is governed by the following rules:

$$R^j : \text{If } \|x\| \text{ is } \tilde{F}^j \text{ THEN } y \text{ is } \tilde{G}^j, j = 1, 2, 3 \tag{5.1}$$

where $\|x\|$ is the normalised input which is described further in Section 5.3 of this chapter. $\tilde{F}^j$ is defined as an IT2 triangular membership function as shown in Fig. 2.2 and $\tilde{G}^j$ is a singleton with $\underline{\text{SVM}}_{jk}$ as LMF and $\overline{\text{SVM}}_{jk}$ as UMF, $k = 1$, 2, 3, denoting the number of IT2FSVMs in Fig. 2.3. $\underline{\text{SVM}}_{jk}$ and $\overline{\text{SVM}}j_k$ are two SVMs with the output $\underline{Out}_{jk}$ and $\overline{Out}_{jk}$ defined by the following hyperplanes:

$$\underline{Out}_{jk} = \text{sgn}(\underline{\omega}_{jk} \cdot z + \underline{b}_{jk}) = sgn\big(\sum_{i=1}^{N} \alpha_{ijk} y_i K(x_i, x) + \underline{b}_{jk}\big) \tag{5.2}$$

$$\overline{Out}_{jk} = \text{sgn}(\overline{\omega}_{jk} \cdot z + \bar{b}_{jk}) = sgn\big(\sum_{i=1}^{N} \alpha_{ijk} y_i K(x_i, x) + \bar{b}_{jk}\big) \tag{5.3}$$

where $j = 1$, 2, 3 denotes the j-th (lower or upper) SVM in Fig. 2.3 and $k = 1$, 2, 3 denotes the k-th IT2 SVM in Fig. 2.3. Further clarification of Equations 5.2 and 5.3 can be obtained by reviewing the SVM theory in Section 2.2.2 in Chapter 2 of the thesis. The $Output_k$ of the IT2 SVM block k can then be obtained by the defuzzification process outlined in Section 2.2.4 in Chapter 2 of the thesis.

The output of the IT2 SVM blocks are then fed into the rule-based class determiner. The rule-based class determiner was designed in order to maximise the classification accuracy of the IT2FSVM classifier. The output from the 3 SVM blocks are compared and the particular class that has the majority amongst the 3 SVM blocks is assigned the final class output. The system for selecting the final classification output for the IT2FSVM is shown in Table. 5.1. The final class is a whole number between 1 and 3 where “1” represents the seizure-free phase, “2” represents the pre-seizure phase and “3” represents the seizure phase.

5.3 Method

The epilepsy problem has been introduced and discussed previously in Section 4.3.2 in Chapter 4 of the thesis. This is the problem that we attempt to address with the aid of the IT2FSVM. A classifier based on the proposed IT2FSVM structure has been implemented for the classification of the 3 seizure phases with the aid of the feature vectors obtained from the feature extraction method as shown in the Epilepsy Seizure Phase Prediction section found in Chapter 4 of the thesis . The structure of the IT2FSVM consists of 3 IT2 SVM blocks that are used to distinguish between the 3 seizure phases. Fig. 5.1 shows the overall structure of the FSVM classifier which consists of 18 45-input-single-output SVMs (6 for each of the IT2 SVM blocks). The 3 sets of SVMs attempt to distinguish between 3 classes of data stems from the fact that the SVM can only separate between 2 classes at any given time.

There are 3 fuzzy rules for each of the IT2 SVM blocks. The parameters of the triangular membership functions, i.e., p_1 to p_7, as shown in Fig. 2.2 are optimised by the GA in order to influence the shape of the membership functions. The GA optimization is performed to maximise the classification accuracy using 70% of dataset as the training samples. The rest 30% of dataset are used as the test samples. The lower and upper membership functions for SVM Block 1 to 3 after training are shown in Figs. 5.2 to 5.4. The membership grade is represented on the y-axis and the normalised inputs are represented on the x-axis. The

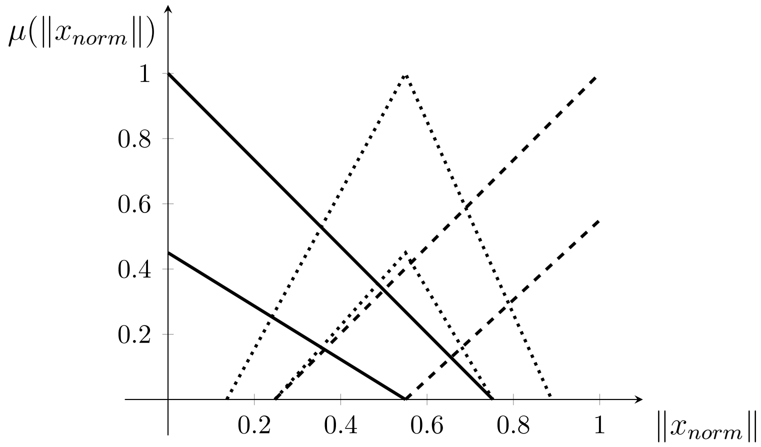

Figure 5.2: Membership functions for SVM Block 1. Dotted line: Membership function for rule 1, Straight Line: Membership function for rule 2, Dashed Line: Membership function for rule 3.

normalised input denoted as x_{norm} is calculated as follows:

$$x_{norm} = \overline{x}_1{}^2 + \overline{x}_2{}^2 + \ldots + \overline{x}_N{}^2 \tag{5.4}$$

where

$$\overline{x}_i = \frac{x_i}{\max(x) - \min(x)}, i = 1, 2, \ldots, N, \tag{5.5}$$

x_i is the i-th element of the vector x, $\min(x)$ and $\max(x)$ denote the minimum and maximum value of the elements in x, respectively.

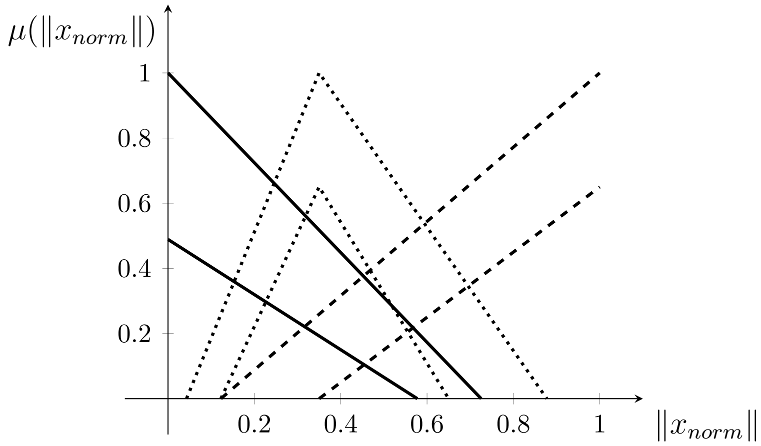

Figure 5.3: Membership functions for SVM Block 2. Dotted line: Membership function for rule 1, Straight Line: Membership function for rule 2, Dashed Line: Membership function for rule 3.

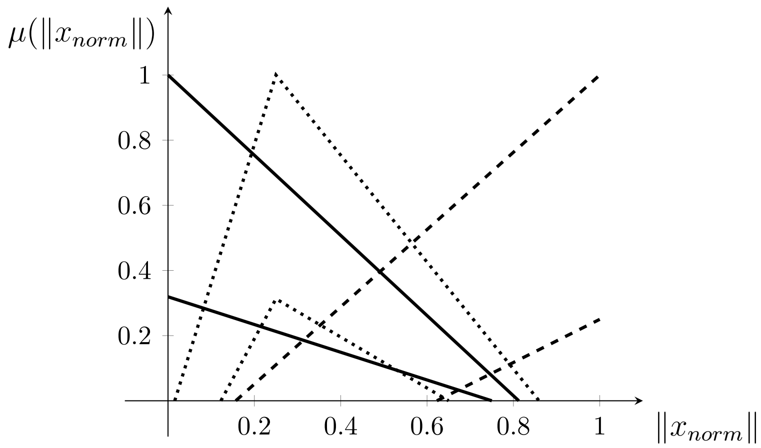

Figure 5.4: Membership functions for SVM Block 3. Dotted line: Membership function for rule 1, Straight Line: Membership function for rule 2, Dashed Line: Membership function for rule 3.

Parameter	Value
Number of Iterations	10
Population Size	20
Selection	Stochastic uniform selection function
Elitism	Yes (Best two chromosomes are passed onto the next generation)
Crossover	Scattered Crossover
Crossover Fraction	0.8
Mutation	Gaussian Mutation
Stopping Criterion	It stops when the weighted average relative change in the best fitness function value over 100 generations is less than or equal to 10^{-6}

Table 5.2: GA Parameters

The simulations that have been conducted with MATLAB. The control parameters of the GA are shown in Table. 5.2. Different combinations of kernel functions are utilised in the SVMs. The optimal combination was chosen based on its ability to maximise the classification accuracy of the classifier. The parameters used for the SVM are as follows: In the IT2 SVM1, there are 6 SVMs used, with all utilizing the RBF kernel function with the width of the RBFs for all 6 of them set to $\sqrt{1/200}$, and the regularization constant $C = 500$; In IT2 SVM2 6 SVMs are used, with the polynomial kernel function applied in all cases and the degree of polynomial set to 2, and $C = 5000$; In IT2 SVM3 the kernel function utilised for all SVMs is the quadratic kernel function with $C = 500$.

In order to obtain an appreciation of the robustness of the proposed classifier, white Gaussian noise with the levels of 0.05, 0.1, 0.2 and 0.5 have been added to the original test dataset. Under these noisy conditions, the simulations were carried out 10 times for each of the noise levels and four statistical factors namely worst, average, best and standard deviation of classification accuracy were calculated. We take these four statistical factors into account since the noisy data is random in nature and drawing conclusions from a single simulation would not accurately evaluate the robustness of the classifier to noise.

5.4 Experimental Results and Analysis

The proposed IT2FSVM classifier is used to classify between the 3 epilepsy seizure phases using the feature vector that has been obtained by the method detailed in Section 4.3.2.1 in Chapter 4 of the thesis. For comparison purposes, 3 traditional classifiers (kNN, naive Bayes and SVM classifiers) are considered. When traditional SVM classifier is considered, they are connected in the classifier structure as shown in Fig. 2.3, i.e., replacing the IT2SVM with the traditional SVM. For the design of the hyperplane, all three traditional SVMs take the RBF kernel with the width of $\sqrt{1/1400}$ and regularization constant $C = 500$.

The classification accuracy with respect to the training dataset for all classifiers is given in Tables 5.3 and 5.4. The tables show the training and testing classification accuracy from the best performed classifiers during the design. They tabulate the worst (among the three classes), best (among the three classes), average (over the three classes) and individual class classification accuracy for both training and testing dataset.

It can be seen from Table. 5.3 that the kNN classifier performs the best in terms of average classification accuracy of 100%. The IT2FSVM classifier comes in the second place with 99.0510% (less than 1% compared with the kNN classifier). This however is not an indication of the kNN being a superior classifier as we see that it suffers from a significant reduction in its average classification performance when exposed to unseen test data with and without noise as seen in column 3 of Table. 5.4 and column 2 of Tables 5.5-5.8. The 100% average training accuracy seen in Table. 5.3 is reduced to 56.6667% in Table.5.4 when the classifier is subjected to the test data. Another significant impact of this is that the kNN has an individual testing classification accuracy of 23.3333% as seen in the 5th column of Table.5.4 when classifying the pre-seizure phase (class 2), this is significant because the accurate classification of the pre-seizure phase is a core objective in addressing the problem of epilepsy seizure phase classification. This would give the patients the advance warning and therefore sufficient time to prepare for the onset of the seizure. The SVM and naive Bayes are ranked third and fourth. Table. 5.4 shows that the IT2FSVM classifier outperforms other classifiers in terms of average classification accuracy for testing dataset. It also shows that the IT2FSVM demonstrates an outstanding generalization ability dealing with unseen data. Compared with other classifiers, the average testing classification accuracies are 10% to 21% higher. The results show that the naive Bayes classifier performs the worst to the testing data and its generalization capability is the poorest. Referring to the worst individual class testing classification accuracy, IT2FSVM can still achieve 70% while other degrade around 23% to 50%.

Tables 5.5 to 5.8 show the testing classification accuracy for the testing data subject to Gaussian noise with the levels of 0.05, 0.1, 0.2 and 0.5. The experiments were repeated 10 times for each classifiers. The "Worst" and the "Best" columns show the worst and best testing individual class classification accuracies among the 10 experiments. The "Mean" and "Std" columns show the mean and standard deviation of the average testing accuracies of the three classes of the 10 experiments. The columns for "Class 1", "Class 2" and "Class 3" show average testing classification accuracy for classes 1 to 3, respectively, of the 10 experiments.

In general, the classification accuracies decreases for all classifiers when the

	Classification Accuracy (%)			
Classifier	Average	Class 1	Class 2	Class 3
1	99.0510	100.000	97.1400	100.0000
2	86.6667	100.0000	90.0000	70.0000
3	100.0000	100.0000	100.0000	100.0000
4	77.1400	90.0000	41.4333	100.0000

Table 5.3: Summary of training samples classification performance for EEG signal with original dataset. Classifier 1: FSVM classifier, classifier 2: traditional SVM classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

noise level increases. In most of the cases, the average testing classification of IT2FSVM and naive Bayes classifiers achieve the best result. However, when it is down to the individual class classification accuracy, especially for higher noise levels (0.1, 0.2 and 0.5), the IT2FSVM performs more robustly with the lowest class classification accuracy of 40% while other classifiers obtain lower class classification accuracies ranging from 15% to 36%. Similar to the comment concerning the kNN and its poor performance in accurately classifying the pre-seizure phase (Class 2), it is important to also note that the nave Bayes classifier exhibits a relatively poor ability to classify the pre-seziure phase as we see that the SVM and IT2FSVM provides superior class classification for the pre-seizure phase in the training, noise-free testing and noise testing of both classifiers. This is a critical difference between these classifiers. The IT2FSVM however achieves a superior overall/average classification accuracy when compared to the SVM and this result shows the superiority and suitability of the IT2FSVM for classifying the three epilepsy seizure phases.

	Classification Accuracy (%)			
Classifier	Average	Class 1	Class 2	Class 3
1	87.7800	100.000	70.0000	93.3300
2	71.1100	90.0000	70.0000	53.333
3	56.6667	96.6700	23.3333	50.0000
4	77.7778	100.0000	33.3333	100.0000

Table 5.4: Summary of testing samples classification performance for EEG signal with original dataset. Classifier 1: FSVM classifier, classifier 2: traditional SVM classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)						
Classifier	Worst	Mean	Best	Std	Class 1	Class 2	Class 3
1	62.2200	66.1100	68.8900	0.0211	8.3000	93.0000	97.0000
2	61.1100	66.2200	68.8900	0.0235	11.1333	96.0000	97.3333
3	56.6700	57.8900	58.8900	0.0176	96.0000	25.0000	52.6700
4	77.7778	78.3333	80.0000	0.7857	99.0000	37.8900	100.0000

Table 5.5: Summary of testing classification performance for EEG signal under dataset subject to noise level of 0.05. Classifier 1: FSVM classifier, classifier 2: traditional SVM classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)						
Classifier	Worst	Mean	Best	Std	Class 1	Class 2	Class 3
1	74.4400	79.4400	85.5600	0.0034	55.3333	66.3333	99.0000
2	66.6700	68.6700	70.0000	0.0126	15.0000	89.0000	98.3333
3	54.4400	56.2200	57.8800	0.0228	92.6700	22.0000	54.0000
4	76.6667	78.8889	82.2222	1.8251	100.0000	33.6667	100.0000

Table 5.6: Summary of testing classification performance for EEG signal under dataset subject to noise level of 0.1. Classifier 1: FSVM classifier, classifier 2: traditional SVM classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)						
Classifier	Worst	Mean	Best	Std	Class 1	Class 2	Class 3
1	73.3300	78.0000	83.3300	0.0384	94.0000	40.6667	99.3300
2	72.2200	74.6700	80.0000	0.0250	27.6667	79.3333	99.3333
3	50.0000	53.3333	55.7800	0.0207	91.6667	21.6667	54.0000
4	76.6667	79.0000	82.2222	1.8898	99.3333	34.3333	100.0000

Table 5.7: Summary of testing samples classification performance for EEG signal under dataset subject to noise level of 0.2. Classifier 1: FSVM classifier, classifier 2: traditional SVM classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

	Classification Accuracy (%)						
Classifier	Worst	Mean	Best	Std	Class 1	Class 2	Class 3
1	73.3300	78.0000	82.2200	0.0295	86.0000	48.0000	100.0000
2	67.7800	68.6700	70.0000	0.0126	36.3333	76.3333	98.3333
3	54.4444	56.6700	58.8900	0.0236	92.6700	23.0000	54.3333
4	75.5556	78.2222	80.0000	1.5585	99.6667	33.6667	100.0000

Table 5.8: Summary of testing classification performance for EEG signal under dataset subject to noise level of 0.5. Classifier 1: FSVM classifier, classifier 2: traditional SVM classifier, classifier 3: K-nearest neighbor classifier, 4: Naive Bayes classifier.

5.5 Conclusion

In this chapter, a novel classification method, IT2FSVM was proposed to use EEG to classify the epileptic seizure from patients with neurological disorder symptoms, where the three epileptic seizure phases seizure-free, pre-seizure and seizure were taken into account. The IT2FSVM merges the SVM and IT2FIS to create a hybrid classifier which attempts to achieve more accurate classification when compared to the traditional classifiers. The simulation results show that the IT2FSVM can achieve more accurate classifications than the traditional kNN, naive Bayes and SVM method do when the classifier is subjected to the original and uncontaminated input data. The input data was then contaminated with noise in order to evaluate the robustness of the proposed IT2FSVM. The validation results show that the proposed IT2FSVM achieve a more significant level of robustness to noisy data when compared to other classification methods. Future research direction will aim to optimise the membership function and the IT2FSVM architectures in order to further improve the overall classification accuracy.

5.6 Discussion - Material Surface Classification

In this section, we discuss the performance of the implemented classifiers in terms of their ability to deal with the material classification problem. The results from best out of the 6 neural network architectures reviewed in Chapter 3 (the tree-structured classifier) and also the VWNN proposed in Chapter 4 which were implemented in an attempt to deal with the material surface classification problem are presented and discussed. The training and testing classification accuracy under the original dataset can be found in Table. 5.9. In the table we have the worst and average classification accuracies for the training and testing datasets. The worst classification accuracy represents the worst individual accuracy amongst the 18 materials whilst the average classification accuracy is obtained by calculating the average individual accuracy of the 18 materials.

In reference to Table. 5.9 we observe a high level of performance in terms of classification accuracy at the different number of feature points used for both classifiers with the lowest average classification being 98.61% recorded by both the tree-structured and VWNN classifiers when 4 feature points are used for classification. In terms of the individual classification accuracy, the lowest classification accuracy is 90%. When comparing both classifiers in terms of the results obtained, we observe that the tree-structured classifiers outperforms the VWNN when using 3 and 5 feature points as the input data for classification with the performance of both classifiers being equal when 4 feature points are used. The testing classification accuracy under a dataset subjected to noise is shown in Table. 5.10. The results show that the tree-structured classifier outperforms the VWNN when 3, 4 and 5 feature points are used as the input into the clas-

[H]

		Classification Accuracy (%)			
		Training		Testing	
#feature points	Classifier	Worst Individual	Avg	Worst Individual	Avg
3	Tree-Structured	97.5000	99.8611	90.0000	99.1667
3	VWNN	100.0000	100.0000	90.0000	98.6111
4	Tree-Structured	100.0000	100.0000	90.0000	98.6111
4	VWNN	100.0000	100.0000	90.0000	98.6111
5	Tree-Structured	100.0000	100.0000	100.0000	100.0000
5	VWNN	100.0000	100.0000	95.0000	99.1667

Table 5.9: Summary of classification performance of the Tree-Structured and VWNN classifiers for Material Classification under noise-free dataset.

sifer with an average classification accuracy of 97.8611%, 97.3611% and 99.7778% when compared to the VWNN which had a classification accuracy of 94.2222%, 91.2222% and 96.3889% respectively for 3, 4 and 5 feature points. In terms of the worst individual classification accuracy, we see that the VWNN performs poorly also as it has a classification accuracy of 0% for one of the classes when 4 feature points are being used. From these results we come to the conclusion that the tree-structured classifier poses a superior level of performance when compared to the VWNN and it is therefore the most effective classifier at dealing with this problem. The tree-structured classifier also has a lower level of complexity whilst offering a higher level of flexibility in implementation.

		Classification Accuracy (%)				
#feature points	Classifier	Worst	Avg	Best	Std	Worst Individual (Average)
3	Tree-Structured	96.3889	97.8611	99.1667	0.9679	80.5000
3	VWNN	91.3889	94.2222	97.2222	2.0916	50.0000
4	Tree-Structured	95.5556	97.3611	99.1667	1.0499	75.5000
4	VWNN	88.6111	91.2222	93.0556	1.4722	0.0000
5	Tree-Structured	99.1667	99.7778	100.0000	0.2869	96.0000
5	VWNN	93.6111	96.3889	98.3333	1.5817	68.0000

Table 5.10: Noise: Summary of classification performance of the Tree-Structured and VWNN classifiers for Material Classification under dataset subject to a noise level of 0.005

5.7 Discussion - Epilepsy Seizure Phase Classification

In this section, we discuss the performance of the implemented classifiers in terms of their ability to deal with the epilepsy seizure phase classification problem. The VWNN classifier in Chapter 4 and IT2FSVM classifier in Chapter 5 are implemented in order to solve this problem. The results obtained from the simulations carried out are presented and discussed. The training and testing classification accuracy under the original dataset can be found in Table. 5.11. In the table we have the worst and average classification accuracies for the training and testing datasets. The worst classification accuracy represents the worst individual accuracy amongst the 3 seizure phases whilst the average classification accuracy is obtained by calculating the average individual classification accuracy of the 3 seizure phases.

Referring to the results in Table. 5.11 we see that the VWNN performs better than the FSVM both with the training stage with an average classification accuracy of 100% compared to that of the IT2FSVM which is 99.0500%. The VWNN also proves to have superior performance when using the testing data with an average classification accuracy of 91.1111% compared to that of the IT2FSVM which is 87.7800%. We also observe that the worst individual accuracy in the testing data for the VWNN is 80% compared to 70% when using the IT2FSVM classifier.

The testing classification accuracy under a dataset that has been subjected to noise at levels of 0.05, 0.1, 0.2 and 0.5 is shown in Table. 5.12. The results show that the VWNN outperforms the IT2FSVM in at all noise levels with an average

[H]

	Classification Accuracy (%)			
	Training		Testing	
Classifier	Worst	Average	Worst	Average
VWNN	100	100	80.0000	91.1111
IT2FSVM	97.1400	99.0500	70.0000	87.7800

Table 5.11: Summary of classification performance of the VWNN and IT2FSVM for EEG signals with original dataset.

classification accuracy of 89.2222%, 86.4444%, 84.4444% and 83.5556% for noise levels of 0.05, 0.1, 0.2 and 0.5 respectively. From the results obtained, we come to the conclusion that the VWNN classifier poses a high level of performance in dealing with this problem when compared to the IT2FSVM. The high level of classification accuracy for the classification between the 3 seizure phases show that it is a suitable classifier to deal with this problem.

	Classification Accuracy (%)				
Classifier	Noise level	Worst	Average	Best	Std
VWNN	0.05	85.5556	89.2222	94.4444	2.9801
IT2FSVM	0.05	62.2200	66.1100	68.8900	0.0211
VWNN	0.10	80.0000	86.4444	92.2222	4.1869
IT2FSVM	0.10	74.4400	79.4400	85.5600	0.0034
VWNN	0.20	78.8889	84.4444	90.0000	3.4978
IT2FSVM	0.20	73.3300	78.0000	83.3300	0.0384
VWNN	0.50	81.1111	83.5556	85.5556	1.5948
IT2FSVM	0.50	73.3300	78.0000	82.2200	0.0295

Table 5.12: Summary of VWNN and IT2FSVM classification performance for EEG signal under dataset subject to noise levels ranging from 0.05 to 0.5.

Chapter 6

Conclusions and Future work

The research that was conducted in this thesis involved an investigation into a number of classification techniques and their application to solve two research problems. The classification techniques introduced were the neural network, variable-weight neural network and interval type-2fuzzy support vector machine with the Naive Bayes and kNN classifiers used for comparison. The novel and existing methods were proposed in order to deal with two particular problems, that of material surface classification and epilepsy seizure phase classification.

- For material surface classification where the input data is obtained with the aid of a tactile sensing robotic finger, research was carried out into 6 well known neural network architectures (one-against-all, weighted one-against-all, binary coded, parallel structured, weighted parallel-structured and tree-structured) which are applied as the classifiers in an attempt to solve the material classification problem. There were 18 household materials which were used as the input. In this approach, the viability of the classifier is justified based on its classification accuracy and also the robustness to the introduction of noise to the input. The performance of the classifiers were then compared to well-known traditional classification methods (kNN and Naive Bayes) with and without noise in the inputs. From the experimental results we see that the parallel-structured, tree-structured and nave Bayes classifiers outperform all the others, the overall best classifier is the tree-structured classifier as it demonstrates a high level of robustness to noise when compared to the others.

- A novel classifier known as the variable weight neural network (VWNN) was proposed in chapter 4. The variable weight neural network allows its weights to be changed based on the inputs to the network. This is done by having two neural networks known as the tuning network and the tuned network. The tuning network produces output weights vectors which consists of all the interconnection weights for the tuned neural network. The tuned neural network then uses the weights obtained from the tuning network to process the input. This gives it the unique ability of being able to adapt to a wide range of inputs into the network and therefore offer better generalization performance when compared to a neural network with fixed weights. The proposed classifier is able to have an improved capability in being able to deal with a wide range of classification problems. The capability of the VWNN is tested by applying it to two real world problems, that of epilepsy seizure phase classification and the material classification problem. Once again the KNN and Nave Bayes classifiers are used as a comparison to the performance of the VWNN. The simulation results show that the VWNN has a superior performance to the traditional classifiers in terms of the classification error and also robustness when the input data is contaminated with noise.

- The third and final significant contribution shown in the thesis involves the merging of interval type-2 fuzzy logic with the support vector machine to propose a novel interval type-2 fuzzy support vector machine (IT2FSVM). The IT2FSVM was applied to the epileptic seizure classification problem where the aim is to classify between three epileptic seizure phases (seizure-free, pre-seizure and seizure). The performance of the IT2FSVM classifier was measured based on its ability to accurately recognize the three epileptic seizure phases. The IT2FSVM classifier shows superior learning capabilities with the original data when compared to other classifiers. The IT2FSVM also shows a high level of robustness to noise when white Gaussian noise is applied to it.

The research conducted in this thesis has been successful in its ability to deal with the two proposed problems (epileptic seizure classification and material classification) as the classifiers proposed in the thesis have been able to show a high level of classification accuracy in the classification involved in both problems.

6.1 Future Work

In this section, some ideas are discussed with the purpose of utilizing them for future work on the research carried out in this thesis.

- For the IT2FSVM discussed in chapter 5, I would investigate different parameters for the classifier to deduce its effect on performance (i.e. in terms of being able to significantly improve the classification accuracy of the classifier). The parameters referred to include the IT2FIS parameters (e.g membership function shape, type-reduction method) and genetic algorithm (GA) parameters (mutation method, no of generations e.t.c) since the GA is used to optimize the shape of the membership function. The SVM parameters (kernel function and regularization constant) were optimized manually via trial and error, a future work would be to utilize the GA or any other optimization method in optimizing the SVM parameters.

- For the material classification problem, further research could be carried out into other methods of obtaining the input data inspired from methods used in the literature review (such as modal analysis, polarimetric imaging or photon scattering) and also different feature extraction methods investigated to obtain feature vectors. In the research conducted, the moments of the distribution used for feature extraction were the mean and variance (1st and 2nd order). Further research could be carried out into using higher other moments such as skewness and kurtosis. The performance would be judged based on their ability to improve the classification accuracy by making it easier for the classifiers to differentiate between classes.

- For the epilepsy seizure phase classification, from the research conducted I noticed that the greatest difficulty was in being able to differentiate between the seizure-free and pre-seizure classes (the EEG pattern of the seizure phase is significantly different from the others and therefore easily separable) and this had a signficant effect on the overall classification accuracy of the classifier. In the research conducted in the thesis, the same feature extraction method has been used for the data obtained from all 3 seizure phases.

Research could be conducted into other signal processing and feature extraction techniques that could be better suited to extracting the distinct features in both classes.

- Tackle a time-series prediction problem (such as financial time-series) by applying the IT2FSVM method proposed in the thesis. In this instance, the SVMs are replaced by SVRs (Support Vector Regressors). This approach can be further improved by implementing a fuzzy kernel approach to combine the existing SVM kernels with the aid of a membership function with a membership grade being attached to the output from each of the kernels. The VWNN can also be utilised in a time-series application.

References

[1] Sally Al-Omar, Walid Kamali, Mohamad Khalil, and Alaa Daher. Classification of eeg signals to detect epilepsy problems. In *Advances in Biomedical Engineering (ICABME), 2013 2nd International Conference on*, pages 5–8. IEEE, 2013.

[2] Sally Al-Omar, Walid Kamali, Mohamad Khalil, and Alaa Daher. Classification of EEG signals to detect epilepsy problems. In *2013 2nd International Conference on Advances in Biomedical Engineering (ICABME)*, pages 5–8. IEEE, 2013.

[3] Frédérique Amor, Sylvain Baillet, Vincent Navarro, Claude Adam, Jacques Martinerie, and Michel Le Van Quyen. Cortical local and long-range synchronization interplay in human absence seizure initiation. *Neuroimage*, **45**[3]:950–962, 2009.

[4] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis – A brief tutorial. *Institute for Signal and information Processing*, 1998.

[5] Gabriel I Barbash and Sherry A Glied. New technology and health care costs-the case of robot-assisted surgery. *New England Journal of*

Medicine, **363**[8]:701–704, 2010.

[6] Raman Bhati, Sarika Jain, Nilesh Maltare, and Durgesh Kumar Mishra. A comparative analysis of different neural networks for face recognition using principal component analysis, wavelets and efficient variable learning rate. In *2010 International Conference on Computer and Communication Technology (ICCCT)*, pages 526–531. IEEE, 2010.

[7] Soudeh Kasiri Bidhendi, Abbas Sarraf Shirazi, Narges Fotoohi, and Mohammad Mehdi Ebadzadeh. Material classification of hyperspectral images using unsupervised fuzzy clustering methods. In *Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, 2007*, pages 619–623. IEEE, 2007.

[8] William M Campbell, Douglas E Sturim, and Douglas A Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, **13**[5]:308–311, 2006.

[9] Oscar Castillo, Patricia Melin, Janusz Kacprzyk, and Witold Pedrycz. Type-2 fuzzy logic: theory and applications. In *Granular Computing, 2007. GRC 2007. IEEE International Conference on*, pages 145–145. IEEE, 2007.

[10] A. E. Cavanna. and F. Monaco. Brain mechanisms of altered conscious states during epileptic seizures. *Nat Rev Neurol*, **5**:267–76, May. 2009.

[11] He Chao and Li Junmin. Adaptive learning control for finite interval tracking based on variable wavelet neural network. In *2013 32nd Chinese Control Conference (CCC)*, pages 3031–3036. IEEE, 2013.

[12] Wanpracha Art Chaovalitwongse, Ya-Ju Fan, and Rajesh C Sachdeo. On the time series k-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **37**[6]:1005–1016, 2007.

[13] Wanpracha Art Chaovalitwongse, Rebecca S Pottenger, Shouyi Wang, Ya-Ju Fan, and Leon D Iasemidis. Pattern-and network-based classification techniques for multichannel medical data signals to improve brain diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **41**[5]:977–988, 2011.

[14] N. Charniya. Neural network based sensor for classification of material type and its surface properties. In *Proceedings of the 2007 IEEE International Joint Conference on Neural Networks*, pages 424–429, Aug. 2007.

[15] Nadir N Charniya and Sanjay V Dudu. Development of intelligent sensor system for classification of material type using neural networks. In *International Conference on Computational Intelligence and Multimedia Applications, 2007*, **1**, pages 174–178. IEEE, 2007.

[16] Damith Suresh Chathuranga, Zhongkui Wang, Yohan Noh, Thrishantha Nanayakkara, and Shinichi Hirai. Robust real time material classification algorithm using soft three axis tactile sensor: Evaluation of the algorithm. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2093–2098. IEEE, 2015.

[17] Wu-Hua Chen and Wei Xing Zheng. Global exponential stability of impulsive neural networks with variable delay: an LMI approach. *IEEE*

Transactions on Circuits and Systems I: Regular Papers, **56**[6]:1248–1259, 2009.

[18] Zheru Chi, Jing Wu, and Hong Yan. Handwritten numeral recognition using self-organizing maps and fuzzy rules. *Pattern Recognition*, **28**[1]:59–66, 1995.

[19] Cheng-Yi Chiang, Nai-Fu Chang, Tung-Chien Chen, Hong-Hui Chen, and Liang-Gee Chen. Seizure prediction based on classification of EEG synchronization patterns with on-line retraining and post-processing scheme. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC*, pages 7564–7569. IEEE, 2011.

[20] HM Chiang and TY Wang. One-against-one fuzzy support vector machine text categorization classifier. In *IEEE International Conference on Industrial Engineering and Engineering Management, 2008.*, pages 1519–1523. IEEE, 2008.

[21] Jen-Tzung Chien. Linear regression based bayesian predictive classification for speech recognition. *IEEE Transactions on Speech and Audio Processing*, **11**[1]:70–79, Jan. 2003.

[22] Hao Dang, Jonathan Weisz, and Peter K Allen. Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5917–5922, 2011.

[23] Sergio Decherchi, Paolo Gastaldo, Ravinder S Dahiya, Maurizio Valle, and Rodolfo Zunino. Tactile-data classification of contact materials using computational intelligence. *IEEE Transactions on Robotics*, **27**[3]:635–639, 2011.

[24] Zümray Dokur and Tamer Ölmez. Heart sound classification using wavelet transform and incremental self-organizing map. *Digital Signal Processing*, **18**[6]:951–959, 2008.

[25] Mehmet Engin. ECG beat classification using neuro-fuzzy network. *Pattern Recognition Letters*, **25**[15]:1715–1722, 2004.

[26] Ergun Erçelebi and Abdulhamit Subaşı. Classification of EEG for epilepsy diagnosis in wavelet domain using artifical neural network and multilinear regression. In *IEEE 14th International Conference on Signal Processing and Communications Applications, 2006*, pages 1–4. IEEE, 2006.

[27] Richard E Fan, Martin O Culjat, Chih-Hung King, Miguel L Franco, Richard Boryk, James W Bisley, Erik Dutson, and Warren S Grundfest. A haptic feedback system for lower-limb prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **16**[3]:270–277, 2008.

[28] Ramon A Felix, Edgar N Sanchez, and Guanrong Chen. Reproducing chaos by variable structure recurrent neural networks. *IEEE transactions on neural networks/a publication of the IEEE Neural Networks Council*, **15**[6]:1450–1457, 2004.

[29] Yan Feng, Wang Jian-mei, and Xu Hai-mei. A variable-weight combination forecasting model based on GM (1, 1) model and RBF neural network. In *2013 International Conference on Measurement, Information and Control (ICMIC)*, **1**, pages 524–528. IEEE, 2013.

[30] R. S. Fisher., W. van Emde Boas., W. Blume., C. Elger., P. Genton., and P. Lee. Epileptic seizures and epilepsy: definitions proposed by the international league against epilepsy (ILAE) and the international bureau for epilepsy (IBE). *Epilepsia*, **46**:22–30, Apr. 2005.

[31] Ham FM. and Kostanic I. Principles of neurocomputing for science and engineering. 2001.

[32] Stephen J Ford and David M Mckeown Jr. Performance evaluation of multispectral analysis for surface material classification. In *International Geoscience and Remote Sensing Symposium, 1994*, **4**, pages 2112–2116. IEEE, 1994.

[33] Liwen Gao, Xiaohua Lin, Mi Zhong, and Junmin Zeng. A neural network classifier based on prior evolution and iterative approximation used for leaf recognition. In *2010 Sixth International Conference on Natural Computation.*, **2**, pages 1038–1043. IEEE, 2010.

[34] G Geetha and SN Geethalakshmi. Detecting epileptic seizures using electroencephalogram: A new and optimized method for seizure classification using hybrid Extreme learning machine. In *2011 International Conference on Process Automation, Control and Computing (PACC)*, pages 1–6. IEEE, 2011.

[35] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, **4**[1]:1–58, 1992.

[36] Samanwoy Ghosh-Dastidar, Hojjat Adeli, and Nahid Dadmehr. Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. *IEEE Transactions on Biomedical Engineering*, **55**[2]:512–518, 2008.

[37] Xavier Gibert, Vishal M Patel, and Rama Chellappa. Material classification and semantic segmentation of railway track images with deep convolutional neural networks. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 621–625. IEEE, 2015.

[38] A. Gorji., C. Mittag., P. Shahabi., T. Seidenbecher., and H. C. Pape. Seizure-related activity of intralaminar thalamic neurons in a genetic model of absence epilepsy. *Neurobiol Dis*, **43**:266–74, Jun. 2011.

[39] Shlomo Greenberg, Hugo Guterman, and Stanley R Rotman. An unsupervised neural network classifier for automatic aerial image recognition. In *Nineteenth Convention of Electrical and Electronics Engineers in Israel, 1996.*, pages 212–215. IEEE, 1996.

[40] I. Guler. and E. D. Ubeyli. Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. *J Neurosci Methods*, **148**:113–21, Oct. 2005.

[41] Guodong Guo, Stan Z Li, and Kap Luk Chan. Support vector machines for face recognition. *Image and Vision computing*, **19**[9]:631–638, 2001.

[42] MX Han, ZH Xu, and YY Yu. Electric load forecasting using structure variable neural networks. In *1993 IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering*, **5**, pages 355–358. IEEE, 1993.

[43] Yan-You Hao, Zhong-Xian Chi, and De-Qin Yan. Fuzzy support vector machine based on vague sets for credit assessment. In *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, **1**, pages 603–607. IEEE, 2007.

[44] R. Harikumar, Srinath Raghavan, and R. Sukanesh. Genetic algorithm for classification of epilepsy risk levels from EEG signals. In *2005 IEEE Region 10 TENCON 2005*, pages 1–6. IEEE, 2005.

[45] R. Harikumar, R. Sukanesh, and P. Aravindan Bharathi. Genetic algorithm optimization of fuzzy outputs for classification of epilepsy risk levels from EEG signals. In *Asilomar Conference on Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth*, **2**, pages 1585–1589. IEEE, 2004.

[46] R Harikumar, T Vijaykumar, and C Palanisamy. Performance analysis of fuzzy techniques hierarchical aggregation functions decision trees and support vector machine (SVM) for the classification of epilepsy risk

levels from EEG signals. In *2011 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 509–514. IEEE, 2011.

[47] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, **2**[5]:359–366, 1989.

[48] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[49] Qi Huang. Fuzzy support vector machine using particle swarm optimization for high-tech enterprises financing risk assessment. In *2013 Fifth International Conference on Computational and Information Sciences (ICCIS)*, pages 670–673. IEEE, 2013.

[50] Shaker Hanizan Hussain, Ahmad Aljunid Syed, and Yahya Saadiah. A novel hybrid fuzzy-SVM image steganographic model. In *Proceedings of 2010 International Symposium in Information Technology*, pages 1–6. IEEE, 2010.

[51] Milo W Hyde, Stephen C Cain, Jason D Schmidt, and Michael J Havrilla. Material classification of an unknown object using turbulence-degraded polarimetric imagery. *IEEE Transactions on Geoscience and Remote Sensing*, **49**[1]:264–276, 2011.

[52] L. D. Iasemidis., J. C. Sackellares., H. P. Zaveri., and W. J. Williams. Phase space topography and the lyapunov exponent of electrocorticograms in partial seizures. *Brain Topogr*, **2**:187–201, Spring. 1990.

[53] Anil K Jain and Jianchang Mao. A k-nearest neighbor artificial neural network classifier. In *Seattle International Joint Conference on Neural Networks, 1991.*, **2**, pages 515–520. IEEE, 1991.

[54] N. Jamali. Material classification by tactile sensing using surface textures. In *Proceedings of the 2007 IEEE International Conference on Image Processing*, pages 2336–2341, May 2010.

[55] Nawid Jamali and Claude Sammut. Material classification by tactile sensing using surface textures. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2336–2341. IEEE, 2010.

[56] Tahseen Ahmed Jilani and Syed Muhammad Aqil Burney. Multiclass bilateral-weighted fuzzy support vector machine to evaluate financial strength credit rating. In *International Conference on Computer Science and Information Technology, 2008.*, pages 342–348. IEEE, 2008.

[57] Zheng Hongtao Cai Jiling Jiang Jingping. Nonlinear modelling of switched reluctance motor based on variable structure Fuzzy-Neural Networks. *Transactions of China Electrotechnical Society*, **6**:000, 2001.

[58] Chia-Feng Juang and Chia-Ming Chang. Human body posture classification by a neural fuzzy network and home care system application. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **37**[6]:984–994, 2007.

[59] Mohammed Kaleem, A Guergachi, and Sridhar Krishnan. EEG seizure detection and epilepsy diagnosis using a novel variation of empirical

mode decomposition. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4314–4317. IEEE, 2013.

[60] Manisha Kandpal, Varun Kumar Kakar, and Gunjan Verma. Classification of ground vehicles using acoustic signal processing and neural network classifier. In *2013 International Conference on Signal Processing and Communication.*, pages 512–518. IEEE, 2013.

[61] N. Kannathal., M. L. Choo., U. R. Acharya., and P. K. Sadasivan. Entropies for detection of epilepsy in EEG. *Comput Methods Programs Biomed*, **80**:187–94, Dec. 2005.

[62] B. D. Killory., X. Bai., M. Negishi., C. Vega., M. N. Spann., and M. Vestal. Impaired attention and network connectivity in childhood absence epilepsy. *Neuroimage*, **56**:2209–17, Jun. 2011.

[63] Ho-Yon Kim, Kil-Taek Lim, and Yun-Seok Nam. Handwritten numeral string recognition using neural network classifier trained with negative data. In *Eighth International Workshop on Frontiers in Handwriting Recognition, 2002.*, pages 395–400. IEEE, 2002.

[64] C-H King, Martin O Culjat, Miguel L Franco, Catherine E Lewis, Erik P Dutson, Warren S Grundfest, and James W Bisley. Tactile feedback induces reduced grasping force in robot-assisted surgery. *Haptics, IEEE Transactions on*, **2**[2]:103–110, 2009.

[65] GR Kiranmayi and V Udayashankara. Neural network classifier for the detection of epilepsy. In *2013 International conference on Circuits, Controls and Communications.*, pages 1–4. IEEE, 2013.

[66] S.B Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, **31**:249–268, Jul. 2007.

[67] Uttam Kumar, S Kumar Raja, Chiranjit Mukhopadhyay, and TV Ramachandra. Hybrid bayesian classifier for improved classification accuracy. *IEEE Geoscience and Remote Sensing Letters*, **8**[3]:474–477, Nov. 2011.

[68] H. K. Lam, Udeme Ekong, Hongbin Liu, Bo Xiao, Hugo Araujo, Ling, and Kit Yan Chan. A study of neural-network-based classifiers for material classification. *Neurocomputing*, **144**[1]:367–377, November 2014.

[69] H. K. Lam and F. H. F. Leung. Digit and command interpretation for electronic book using neural network and genetic algorithm. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, **34**[6]:2273–2283, Dec. 2004.

[70] H. K. Lam and J Prada. Interpretation of handwritten single-stroke graffiti using support vector machines. *International Journal of Computational Intelligence and Applications*, **8**[04]:369–393, Dec. 2009.

[71] K. F. Leung, F. H. F Leung, H. K. Lam, and S. H. Ling. On interpretation of graffiti digits and characters for eBooks: neural-fuzzy network and genetic algorithm approach. *IEEE Trans. on Industrial Electronics*, **51**[2]:464–471, Apr. 2004.

[72] K. F. Leung, Frank H. F. Leung, H. K. Lam, and Peter K. S. Tam. Neural fuzzy fetwork and genetic algorithm approach for cantonese speech command recognition. In *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, **1**, pages 208–213. IEEE, 2003.

[73] Huiling Li, Chunming Li, and Wei Wang. Artificial neural network classifier design using genetic algorithm and wavelet transform in fault diagnosis. In *2010 2nd International Conference on Information Engineering and Computer Science.*, pages 1–4. IEEE, 2010.

[74] Jianming Li, Shuguang Huang, Ongsheng He, and Kunming Qian. Image classification based on fuzzy support vector machine. In *International Symposium on Computational Intelligence and Design, 2008.*, **1**, pages 68–71. IEEE, 2008.

[75] Lei Li, Wen-Yan Ding, and Jin-Yan Li. A novel robustness image watermarking scheme based on fuzzy support vector machine. In *2010 3rd IEEE International Conference on Computer Science and Information Technology.*, **6**, pages 533–537. IEEE, 2010.

[76] Lei Li, Zhi-ping GAo, and Wen-yan Din. Fuzzy multi-class support vector machine based on binary tree in network intrusion detection. In *2010 International Conference on Electrical and Control Engineering*, pages 1043–1046. IEEE, 2010.

[77] Xue-bin Li and Xiao-Ling Yu. Influence of sample size on prediction of animal phenotype value using back-propagation artificial neural network

with variable hidden neurons. In *International Conference on Computational Intelligence and Software Engineering, 2009*, pages 1–4. IEEE, 2009.

[78] Chin-Teng Lin, Chang-Mao Yeh, Sheng-Fu Liang, Jen-Feng Chung, and Nimit Kumar. Support-vector-based fuzzy neural network for pattern classification. *IEEE Transactions on Fuzzy Systems*, **14**[1]:31–41, 2006.

[79] Chun-Fu Lin and Sheng-De Wang. Fuzzy support vector machines. *IEEE Transactions on Neural Networks*, **13**[2]:464–471, 2002.

[80] S.H Lin, Frank H.F Leung, and H.K Lam. Genetic algorithm based variable-structure neural network and its industrial application. In *30th Annual Conference of IEEE Industrial Electronics Society, 2004*, **2**, pages 1273–1278. IEEE, 2004.

[81] S.H Ling, H.K Lam, Frank H.F Leung, and Y.S Lee. A genetic algorithm based variable structure neural network. In *The 29th Annual Conference of the IEEE Industrial Electronics Society, 2003*, **1**, pages 436–441. IEEE, 2003.

[82] B. Litt. and J. Echauz. Prediction of epileptic seizures. *Lancet Neurol*, **1**:22–30, May. 2002.

[83] Guoping P Liu, Visakan Kadirkamanathan, Stephen Billings, et al. Variable neural networks for adaptive control of nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **29**[1]:34–43, 1999.

[84] Hongbin Liu, Xiaojing Song, Joao Bimbo, Lakmal Seneviratne, and Kaspar Althoefer. Surface material recognition through haptic exploration using an intelligent contact sensing finger. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 52–57. IEEE, 2012.

[85] Hongbin Liu, Xiaojing Song, Thrishantha Nanayakkara, Lakmal D Seneviratne, and Kaspar Althoefer. A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor. In *Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1410–1415. IEEE, 2012.

[86] Yi-Hung Liu and Yen-Ting Chen. Face recognition using total margin-based adaptive fuzzy support vector machines. *IEEE Transactions on Neural Networks*, **18**[1]:178–192, 2007.

[87] ZJ Liu, CY Wang, Z Niu, and AX Liu. Evolving multi-spectral neural network classifier using a genetic algorithm. In *2002 IEEE International Geoscience and Remote Sensing Symposium, 2002.*, **6**, pages 3480–3482. Ieee, 2002.

[88] Gian Luca Marcialis and Fabio Roli. Fusion of multiple fingerprint matchers by single-layer perceptron with class-separation loss function. *Pattern Recognition Letters*, **26**[12]:1830–1839, Sep. 2005.

[89] H. K. Meeren., J. P. Pijn., E. L. Van Luijtelaar., A. M. Coenen., and F. H. Lopes da Silva. Cortical focus drives widespread

corticothalamic networks during spontaneous absence seizures in rats. *J Neurosci*, **22**:1480–95, Feb. 2002.

[90] Hassen Mekki and Mohamed Chtourou. Variable structure neural networks for online identification of continuous-time dynamical systems using evolutionary artificial potential fields. In *2012 9th International Multi-Conference on Systems, Signals and Devices (SSD)*, pages 1–6. IEEE, 2012.

[91] Jerry M Mendel. Type-2 fuzzy sets and systems: An overview. *IEEE Computational Intelligence Magazine*, **2**[1]:20–29, 2007.

[92] Neri Merhav and Yariv Ephraim. A bayesian classification approach with application to speech recognition. *Signal Processing, IEEE Transactions on*, **39**[10]:2157–2166, 1991.

[93] Dieter Merkl. Text classification with self-organizing maps: Some lessons learned. *Neurocomputing*, **21**[1]:61–77, 1998.

[94] M.M Mirbagheri, K. Badie, R.M Golpayegani, and M. Amir Ahmadi. A neural network approach to EEG classification for the purpose of differential diagnosis between epilepsy and normal EEG states. In *14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2649–2650. IEEE, 1992.

[95] Seyed Alireza Mohseni and Ai Hui Tan. Optimization of neural networks using variable structure systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **42**[6]:1645–1653, 2012.

[96] Nilesh N Karnik and Jerry M Mendel. Operations on type-2 fuzzy sets. *Fuzzy sets and systems*, **122**[2]:327–348, 2001.

[97] M. Niknazar., S. R. Mousavi., S. Motaghi., A. Dehghani., B. Vosoughi Vahdat., and M. B. Shamsollahi. A unified approach for detection of induced epileptic seizures in rats using ECoG signals. *Epilepsy Behav*, **27**:355–64, Mar. 2013.

[98] Nuryani Nuryani, Sai Ho Ling, and Hung T Nguyen. Hybrid particle swarm-based fuzzy support vector machine for hypoglycemia detection. In *2012 IEEE International Conference on Fuzzy Systems.*, pages 1–6. IEEE, 2012.

[99] Joseph E ODoherty, Mikhail A Lebedev, Peter J Ifft, Katie Z Zhuang, Solaiman Shokur, Hannes Bleuler, and Miguel A. L. Nicolelis. Active tactile exploration using a brain-machine-brain interface. *Nature*, **479**[7372]:228–231, 2011.

[100] Allison M Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, **31**[6]:499–508, 2004.

[101] Ryuta Ozawa, Ji-Hun Bae, and Suguru Arimoto. Multi-fingered dynamic blind grasping with tactile feedback in a horizontal plane. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1006–1011, 2006.

[102] Yüksel Özbay, Rahime Ceylan, and Bekir Karlik. A fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Computers in Biology and Medicine*, **36**[4]:376–388, 2006.

[103] R. Panda, P.S Khobragade, P.D Jambhule, S.N Jengthe, P.R Pal, and T.K Gandhi. Classification of EEG signal using wavelet transform and support vector machine for epileptic seizure detection. In *IEEE Transactions on Systems in Medicine and Biology (ICSMB*, pages 405–408. IEEE, 2010.

[104] Clement C.C Pang, Adrian R.M Upton, Glenn Shine, and Markad V. Kamath. A comparison of algorithms for detection of spikes in the electroencephalogram. *IEEE Transactions on Biomedical Engineering*, **50**[4]:521–526, 2003.

[105] Li Peng, Chen Jie, Cai Tao, and Liu Guoping. On-line identification of fuel cell model with variable neural network. In *2010 29th Chinese Control Conference (CCC)*, pages 1417–1421. IEEE, 2010.

[106] Sunil Kumar Prabhakar and Harikumar Rajaguru. A different approach to epilepsy risk level classification utilizing various distance measures as post classifiers. In *2015 8th Biomedical Engineering International Conference (BMEiCON)*, pages 1–5. IEEE, 2015.

[107] Sunil Kumar Prabhakar and Harikumar Rajaguru. Morphological operator based feature extraction technique along with suitable post classifiers for epilepsy risk level classification. In *2015 International Confer-*

ence on Intelligent Informatics and Biomedical Sciences (ICIIBMS), pages 446–451. IEEE, 2015.

[108] J. Pradeep, E. Srinivasan, and S. Himavathi. Neural network based handwritten character recognition system without feature extraction. In *Proceedings of the International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pages 40–44, Jan. 2011.

[109] Luis Rodríguez-Cobo, P Beatriz García-Allende, Adolfo Cobo, José Miguel López-Higuera, and Olga M Conde. Raw material classification by means of hyperspectral imaging and hierarchical temporal memories. *Sensors Journal, IEEE*, **12**[9]:2767–2775, 2012.

[110] Osvaldo A Rosso, Alexandre Mendes, Regina Berretta, John A Rostas, Mick Hunter, and Pablo Moscato. Distinguishing childhood absence epilepsy patients from controls by the analysis of their background brain electrical activity (ii): A combinatorial optimization approach for electrode selection. *Journal of neuroscience methods*, **181**[2]:257–267, 2009.

[111] Fernandez Santillana, Carlos Delgado-Mata, and Ramiro Velazquez. Training a single-layer perceptron for an approximate edge detection on a digital image. In *Proceedings of the International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 189–193, Nov. 2011.

[112] M. Iqbal Saripan, Mohd Saad, Wira Hidayat, Suhairul Hashim, ATA Rahman, Kevin Wells, David Bradley, et al. Analysis of

photon scattering trends for material classification using artificial neural network models. *IEEE Transactions on Nuclear Science*, **60**[2]:515–519, 2013.

[113] KEYONG SHAO, HONGYU GAO, XIANLI YU, YUANYUAN YANG, HUIZHEN ZHANG, AND SHENG LIU. Neural networks variable structure control for nonlinear time-delay systems based on robust control. In *The Sixth World Congress on Intelligent Control and Automation, 2006*, **1**, pages 2461–2464. IEEE, 2006.

[114] ZHAN SHI AND JINGLU HU. Local linear discriminant analysis with composite kernel for face recognition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–5. IEEE, 2012.

[115] ALISTAIR SHILTON AND DANIEL TH LAI. Iterative fuzzy support vector machine classification. In *IEEE International Fuzzy Systems Conference, 2007.*, pages 1–6. IEEE, 2007.

[116] XIAOJING SONG, HONGBIN LIU, KASPAR ALTHOEFER, THRISHANTHA NANAYAKKARA, AND LAKMAL D SENEVIRATNE. Efficient break-away friction ratio and slip prediction based on haptic surface exploration. *IEEE Transactions on Robotics*, 2013, DOI: 10.1109/TRO.2013.2279630.

[117] XIAOJING SONG, HONGBIN LIU, JOAO BIMBO, KASPAR ALTHOEFER, AND LAKMAL D SENEVIRATNE. A novel dynamic slip prediction and compensation approach based on haptic surface exploration. In *Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*, pages 4511–4516. IEEE, 2012.

[118] Octavian Stan and Edward W Kamen. New block recursive mlp training algorithms using the levenberg-marquardt algorithm. In *Neural Networks, 1999. IJCNN'99. International Joint Conference on*, **3**, pages 1672–1677. IEEE, 1999.

[119] R Sukanesh and R Harikumar. Minimum relative entropy (mre) method for fuzzy based classification of epilepsy risk levels from eeg signals. In *Biomedical and Pharmaceutical Engineering, 2006. ICBPE 2006. International Conference on*, pages 93–98. IEEE, 2006.

[120] R. Sukanesh and R. Harikumar. Minimum relative entropy (MRE) method for fuzzy based classification of epilepsy risk levels from EEG signals. In *International Conference on Biomedical and Pharmaceutical Engineering, 2006. ICBPE 2006.*, pages 93–98. IEEE, 2006.

[121] R Sukanesh and R Harikumar. A structured soft (max-min) decision trees for patient specific fuzzy classifier in the classification of epilepsy risk levels from eeg signals. In *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, **1**, pages 435–442. IEEE, 2007.

[122] R. Sukanesh and R. Harikumar. A structured soft (max-min) decision trees for patient specific fuzzy classifier in the classification of epilepsy risk levels from EEG signals. In *International Conference on Computational Intelligence and Multimedia Applications, 2007*, **1**, pages 435–442. IEEE, 2007.

[123] R. Sukanesh and R. Harikumar. Fuzzy techniques and hierarchical aggregation functions decision trees for the classification of epilepsy risk levels from EEG signals. In *TENCON 2008-2008 IEEE Region 10 Conference*, pages 1–6. IEEE, 2008.

[124] M.H Tan, F. Mat, N.L Tajul Lile, S. Yaacob, et al. Classification of materials by modal analysis and neural network. In *2011 International Conference on Information Technology and Multimedia (ICIM)*, pages 1–5. IEEE, 2011.

[125] Hao Tang and Liang-sheng Qu. Fuzzy support vector machine with a new fuzzy membership function for pattern classification. In *2008 International Conference on Machine Learning and Cybernetics*, **2**, pages 768–773. IEEE, 2008.

[126] Tadanari Taniguchi, Kazuo Tanaka, Hiroshi Ohtake, and Hua O Wang. Model construction, rule reduction, and robust compensation for generalized form of takagi-sugeno fuzzy systems. *Fuzzy Systems, IEEE Transactions on*, **9**[4]:525–538, 2001.

[127] Alberto Tellaeche, Xavier P Burgos-Artizzu, Gonzalo Pajares, and Angela Ribeiro. A vision-based method for weeds identification through the Bayesian decision theory. *Pattern Recognition*, **41**[2]:521–530, Feb 2008.

[128] V. Thilak. Material classification using passive polametric imagery. In *Proceedings of the 2007 IEEE International Conference on Image Processing*, pages 121–124, Sept. 2007.

[129] Vimal Thilak, Charles D. Creusere, and David G. Voelz. Material classification using passive polarimetric imagery. In *IEEE International Conference on Image Processing, 2007*, **4**, pages IV–121. IEEE, 2007.

[130] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, **2**:45–66, 2002.

[131] E. D. Ubeyli. Automatic detection of electroencephalographic changes using adaptive neuro-fuzzy inference system employing Lyapunov exponents. *Expert Systems with Applications*, **36**:9031–9038, Jul. 2009.

[132] Ozge Uncu and IB Turksen. Discrete interval type 2 fuzzy system models using uncertainty in learning parameters. *IEEE Transactions on Fuzzy Systems*, **15**[1]:90–106, 2007.

[133] Aditya Vailaya and Abhishek Jain. Incremental learning for bayesian classification of images. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, **2**, pages 585–589. IEEE, 1999.

[134] O. A. J. Van der Meijden and M. P. Schijven. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. *Surgical endoscopy*, **23**[6]:1180–1190, 2009.

[135] S. Vani and G.R Suresh. Performance analysis of lifting based DWT and MLPNN for epilepsy seizure from EEG. In *2013 International Conference on Human Computer Interactions (ICHCI)*, pages 1–7. IEEE, 2013.

[136] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, **73**[2]:185–214, 2008.

[137] Yeqin Wang, Hui Wang, and Lihong Mo. Research on recognition of wood texture based on integrated neural network classifier. In *2010 International Conference on Intelligent Control and Information Processing.*, pages 512–515. IEEE, 2010.

[138] Yongqiao Wang, Shouyang Wang, and Kin Keung Lai. A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems*, **13**[6]:820–831, 2005.

[139] Chongming Wu, Xiaodan Wang, Dongying Bai, and Hongda Zhang. Fast incremental learning algorithm of SVM on KKT conditions. In *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, **1**, pages 551–554. IEEE, 2009.

[140] Dongrui Wu and Jerry M Mendel. Enhanced Karnik–Mendel algorithms. *IEEE Transactions on Fuzzy Systems*, **17**[4]:923–934, 2009.

[141] Huimin Xiao, Chunyi Su, and Wenfang Xie. Variable structure control based on the fuzzy neural networks. In *Annual meeting of the North American Fuzzy Information Processing Society, 2006.*, pages 206–210. IEEE, 2006.

[142] Sheng-Wu Xiong, Hong-Bing Liu, and Xiao-Xiao Niu. Fuzzy support vector machines based on FCM clustering. In *Proceedings of 2005*

International Conference on Machine Learning and Cybernetics., **5**, pages 2608–2613. IEEE, 2005.

[143] Sibel Yaman, Jason Pelecanos, and Weizhong Zhu. Unifying PLDA and polynomial kernel SVMs. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7698–7701. IEEE, 2013.

[144] Fan Yan-qin, Yang You-long, and Qin Yang-sen. Credit scoring model based on pca and improved tree augmented bayesian classification. In *Information and Communications Technologies (IETICT 2013), IET International Conference on*, pages 169–175. IET, 2013.

[145] Zhaohui Yuan, Lihong Huang, Dewen Hu, and Bingwen Liu. Convergence of nonautonomous Cohen-Grossberg-Type neural networks with variable delays. *IEEE Transactions on Neural Networks*, **19**[1]:140–147, 2008.

[146] Igor Zelič, Igor Kononenko, Nada Lavrač, and Vanja Vuga. Induction of decision trees and bayesian classification applied to diagnosis of sport injuries. *Journal of Medical Systems*, **21**[6]:429–444, 1997.

[147] Guoqiang Peter Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **30**[4]:451–462, Nov. 2000.

[148] Hao Zhang, Alexander C Berg, Michael Maire, and Jitendra Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, **2**, pages 2126–2136, 2006.

[149] Jiao Zhang and Yun Yang. An improved BP algorithm for classification of architectural ceramic material. In *2010 International Conference on Computer Design and Applications (ICCDA)*, **2**, pages V2–447. IEEE, 2010.

[150] Jiye Zhang. Globally exponential stability of neural networks with variable delays. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **50**[2]:288–290, 2003.

[151] Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *Proceedings of the IEEE International Conference on Granular Computing*, **2**, pages 718–721, 2005.

[152] Yan Zhang and Bin Yu. Face recognition using combined non-negative principal component analysis and linear discriminant analysis. In *Proceedings of the 2013 IEEE International Congress on Signal and Image Processing*, pages 758–762. IEEE, 2013.

[153] Meng-Meng Zhou, Lei Li, and Yan-Ling Lu. Fuzzy support vector machine based on density with dual membership. In *2009 International Conference on Machine Learning and Cybernetics.*, **2**, pages 674–678. IEEE, 2009.

[154] Wei Zuo and Lilong Cai. A new iterative learning controller using variable structure fourier neural network. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **40**[2]:458–468, 2010.

[155] Jozef Zurada. Could decision trees improve the classification accuracy and interpretability of loan granting decisions? In *Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS)*, pages 1–9, Jan. 2010.